AWS IoT Core

Guia do desenvolvedor

AWS IoT Core: Guia do desenvolvedor

Copyright © Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

As marcas comerciais e imagens de marcas da Amazon não podem ser usadas no contexto de nenhum produto ou serviço que não seja da Amazon, nem de qualquer maneira que possa gerar confusão entre os clientes ou que deprecie ou desprestigie a Amazon. Todas as outras marcas comerciais que não pertencem à Amazon pertencem a seus respectivos proprietários, que podem ou não ser afiliados, patrocinados pela Amazon ou ter conexão com ela.

Table of Contents

O que é o AWS IoT? ... 1
- Como seus dispositivos e aplicativos acessamAWS IoT ... 1
- O queAWS IoTpode fazer ... 2
  - IoT na indústria ... 2
  - IoT na automação residencial ... 3
- Como o AWS IoT funciona ... 3
  - O universo do IoT ... 3
  - AWS IoTVisão geral dos serviços ... 6
  - Serviços da AWS IoT Core ... 9
- Saiba mais sobre o AWS IoT ... 12
  - Recursos de treinamento paraAWS IoT ... 12
  - AWS IoTrecursos e guias ... 12
  - AWS IoTnas redes sociais ... 13
  - AWSserviços usados peloAWS IoT CoreMecanismo de regras ... 13
  - Protocolos de comunicação suportados porAWS IoT Core ... 14
- Novidades doAWS IoTconsole ... 14
  - Legenda ... 17

Conceitos básicos do AWS IoT Core ... 18
- Connect o primeiro dispositivo aoAWS IoT Core ... 18
- Configurar seuConta da AWS ... 19
  - Cadastrar-se em uma Conta da AWS ... 19
  - Crie um usuário e conceda permissões ... 20
  - Abrir oAWS IoTconsole ... 21
- Experimente oAWS IoT Coretutorial interativo ... 21
  - Conectar dispositivos IoT ... 22
  - Salvando o estado do dispositivo offline ... 23
  - Encaminhamento de dados do dispositivo para serviços ... 23
- Experimente oAWS IoTConexão rápida ... 24
  - Etapa 1. Inicie o tutorial ... 25
  - Etapa 2. Criar um objeto de coisa ... 27
  - Etapa 3. Faça download de arquivos para o dispositivo ... 29
  - Etapa 4. Executar a amostra ... 31
  - Etapa 5. Explorar ainda mais ... 33
  - Testando a conectividade com o endpoint de dados do dispositivo ... 34
- Explorar oAWS IoT Coreserviços em hands-on tutorial ... 38
  - Qual opção de dispositivo é a melhor para você? ... 39
  - CriarAWS IoTrecursos ... 39
  - Configurar o dispositivo ... 43
- Visualizar mensagens MQTT com o cliente MQTT do AWS IoT ... 66
  - Visualizar mensagens MQTT no cliente MQTT ... 67
  - Publicar mensagens MQTT do cliente MQTT ... 69

Como conectar-se ao AWS IoT Core ... 71
- AWS IoT Core- endpoints do plano de controle do ... 71
- AWS IoTendpoints de dispositivo ... 72
- AWS IoT Corepara gateways e dispositivos LoRaWAN ... 73
- Conexão aoAWS IoT CoreEndpoints de serviço da ... 73
  - AWS CLI para AWS IoT Core ... 74
  - AWS SDKs ... 74
  - SDKs móveis do AWS ... 78
  - APIs REST doAWS IoT Coreserviços ... 79
- Conectar dispositivos aoAWS IoT ... 79
  - AWS IoTdados do dispositivo e endpoints de serviço ... 80
  - SDKs de dispositivo da AWS IoT ... 81
  - Protocolos de comunicação de ... 83

Tópicos MQTT ... 99
endpoints configuráveis ... 117
Conexão aoAWS IoTEndpoints do FIPS ... 123
AWS IoT Core- endpoints do plano de controle ... 123
AWS IoT Core- endpoints do plano de dados ... 123
AWS IoT Device Management- endpoints de dados de trabalhos ... 123
AWS IoT Device Management- Endpoints do Fleet Hub ... 124
AWS IoT Device Management- endpoints de encapsulamento seguro ... 124
Tutoriais do AWS IoT ... 125
Criando demonstrações com oAWS IoTCliente de dispositivo ... 125
Pré-requisitos para criar demonstrações com oAWS IoTCliente de dispositivo ... 126
Preparando seus dispositivos para oAWS IoTCliente do dispositivo ... 127
Instalar e configurar oAWS IoTCliente do dispositivo ... 137
Demonstrar comunicação de mensagens MQTT com oAWS IoTCliente do dispositivo ... 146
Demonstrar ações remotas (trabalhos) com oAWS IoTCliente do dispositivo ... 160
Liberar ... 169
Construindo soluções com oAWS IoTSDKs de dispositivo da ... 176
Comece a criar soluções com oAWS IoTSDKs de dispositivo da ... 176
Conectar um dispositivo àAWS IoT Coreusando oAWS IoTSDK de dispositivo da ... 177
CriarAWS IoTregras para rotear dados do dispositivo para outros serviços ... 193
Mantendo o estado do dispositivo enquanto o dispositivo está offline com Sombras de dispositivos ... 223
Criação de um autorizador personalizado paraAWS IoT Core ... 243
Monitoramento da umidade do soloAWS IoTe Raspberry Pi ... 255
Gerenciamento de dispositivos com o AWS IoT ... 265
Como gerenciar coisas com o registro ... 265
Criar uma coisa ... 266
Listar coisas ... 266
Descreva coisas ... 268
Atualizar uma coisa ... 268
Excluir uma coisa ... 268
Anexar uma principal a uma coisa ... 268
Desanexar uma principal de uma coisa ... 269
Tipos de coisa ... 269
Criar um tipo de coisa ... 269
Listar tipos de coisa ... 270
Descrever um tipo de coisa ... 270
Associar um tipo de coisa a uma coisa ... 270
Reprovar um tipo de coisa ... 271
Excluir um tipo de coisa ... 272
Grupos de coisas estáticas ... 272
Criar um grupo de coisas estáticas ... 273
Descrever um grupo de coisas ... 274
Adicionar uma coisa a um grupo de coisas estáticas ... 275
Remover uma coisa de um grupo de coisas estáticas ... 275
Listar coisas em um grupo de coisas ... 275
Listar grupos de coisas ... 276
Listar grupos para uma coisa ... 277
Atualizar um grupo de coisas estáticas ... 278
Excluir um grupo de coisas ... 278
Anexar uma política a um grupo de coisas estáticas ... 278
Desanexar uma política de um grupo de coisas estáticas ... 279
Listar as políticas anexadas a um grupo de coisas estáticas ... 279
Listar os grupos para uma política ... 280
Obter políticas efetivas para uma coisa ... 280
Testar a autorização de ações MQTT ... 281
Grupos de coisas dinâmicas ... 282

Criar grupo de coisas dinâmicas 283
Descrever um grupo de coisas dinâmicas 283
Atualizar um grupo de coisas dinâmicas 284
Excluir um grupo de coisas dinâmicas 285
Limitações e conflitos 285
Marcar recursos do AWS IoT 288
Conceitos básicos de tags 288
Restrições e limitações de tags 289
Utilização de tags com políticas do IAM 289
Grupos de faturamento 291
Visualizar alocação de custos e dados de uso 291
Segurança 293
Segurança em AWS IoT 293
Autenticação 294
Treinamento e certificação da AWS 294
Visão geral do certificado X.509 294
Autenticação do servidor 295
Autenticação de cliente 297
Autenticação personalizada 319
Autorização 330
Treinamento e certificação da AWS 332
Políticas do AWS IoT Core 332
Autorizar chamadas diretas paraAWSUsar serviços doAWS IoT CoreProvedor de credenciais 375
Acesso entre contas com o IAM 379
Proteção de dados 380
Criptografia de dados no AWS IoT 381
Segurança de transporte no AWS IoT 381
Criptografia de dados 382
Identity and Access Management 383
Público 384
Autenticar com identidades do IAM 384
Gerenciamento do acesso usando políticas 386
Como o AWS IoT funciona com o IAM 388
Exemplos de políticas baseadas em identidade 404
Políticas gerenciadas pela AWS 407
Solução de problemas 417
Registro em log e monitoramento 420
Ferramentas de monitoramento 420
Validação de conformidade 421
Resiliência 422
Usar o AWS IoT Core com VPC endpoints 422
Criar VPC endpoints para o AWS IoT Core 423
Controle do acesso aoAWS IoT CoreVPC endpoints 423
Limitações dos endpoints da VPC 424
Dimensionando endpoints da VPC com IoT Core 424
Uso de domínios personalizados com VPC endpoints 424
Disponibilidade de VPC endpoints paraAWS IoT Core 424
Segurança da infraestrutura 425
monitoramento de segurança 425
Práticas recomendadas de segurança 425
Proteger conexões MQTT no AWS IoT 425
Manter o relógio do dispositivo sincronizado 427
Validar o certificado do servidor 428
Usar uma identidade única por dispositivo 428
Use um segundoRegião da AWScomo backup 428
Usar provisionamento just-in-time 429
Permissões para executarAWS IoTTestes Device Advisor 429

Prevenção contra o problema entre serviços confusos para o Device Advisor 430
Treinamento e certificação da AWS 431
Monitorar o AWS IoT 432
Configurar registro em log da AWS IoT 433
Configurar a função e a política de registro em log 433
Configurar o registro em log padrão no AWS IoT (console) 434
Configurar registro em log padrão no AWS IoT (CLI) 435
Configurar registro em log de recursos específicos no AWS IoT (CLI) 437
Níveis de log 438
MonitorAWS IoTalarmes e métricas usando o Amazon CloudWatch 439
Usar métricas do AWS IoT 439
Criar CloudWatch Alarmes noAWS IoT 440
Métricas e dimensões do AWS IoT 443
MonitorAWS IoTusando CloudWatch Logs 454
Visualizar oAWS IoTlogs no CloudWatch console 454
CloudWatchAWS IoTEntradas de log do do 455
Registre em log as chamadas à API do AWS IoT usando o AWS CloudTrail. 475
Informações do AWS IoT no CloudTrail 475
Noções básicas sobre entradas de arquivos de log do AWS IoT 476
Regras 478
Concessão de umAWS IoTgoverne o acesso necessário 479
Transmitir permissões de função 480
Criar uma regra do AWS IoT 481
Visualizar as regras 485
Excluir uma regra 485
Ações de regra do AWS IoT 485
Apache Kafka 487
Alarmes do CloudWatch 495
CloudWatch Logs 496
Métricas do CloudWatch 497
DynamoDB 499
DynamoDBv2 501
Elasticsearch 503
HTTP 504
IoT Analytics 531
IoT Events 533
IoT SiteWise 534
Kinesis Data Firehose 538
Kinesis Data Streams 540
Lambda 542
OpenSearch 544
Republish 546
S3 547
Salesforce IoT 549
SNS 549
SQS 551
Step Functions 553
Timestream 554
Solucionar problemas de uma regra 559
Como acessar recursos entre contas usandoAWS IoTregras 559
Pré-requisitos 559
Configuração entre contas para o Amazon SQS 559
Configuração entre contas do Amazon SNS 561
Configuração entre contas do Amazon S3 562
Configuração entre contas paraAWS Lambda 563
Tratamento de erros (ação de erro) 565
Formato da mensagem de ação de erro 565

Exemplo de ação de erro ... 566
Reduzir custos do sistema de mensagens com a Ingestão básica ... 567
Usar a Ingestão básica ... 567
Referência SQL do AWS IoT ... 568
Cláusula SELECT ... 569
Cláusula FROM ... 570
Cláusula WHERE ... 571
Tipos de dados ... 572
Operadores ... 575
Funções ... 581
Literais ... 624
Declarações de caso ... 625
Extensões JSON ... 626
Modelos de substituição ... 627
Consultas de objeto aninhado ... 629
Cargas binárias ... 630
Versões do SQL ... 631
Serviço Device Shadow ... 633
Usar shadows ... 633
Optar por usar sombras nomeadas ou sem nome ... 633
Acessar sombras ... 634
Usar sombras em dispositivos, aplicativos e outros serviços na nuvem ... 634
Ordem das mensagens ... 635
Reduzir mensagens de shadow ... 636
Usar sombras em dispositivos ... 637
Inicializar o dispositivo na primeira conexão ao AWS IoT ... 638
Processar mensagens enquanto o dispositivo está conectado ao AWS IoT ... 639
Processar mensagens quando o dispositivo se reconecta ao AWS IoT ... 640
Usar sombras em aplicativos e serviços ... 640
Inicializar o aplicativo ou serviço na conexão com o AWS IoT ... 641
Processar alterações de estado enquanto o aplicativo ou serviço está conectado ao AWS IoT ... 641
Detectar se um dispositivo está conectado ... 641
Simulando comunicações de serviço Device Shadow ... 643
Configurar a simulação ... 643
Inicializar o dispositivo ... 643
Enviar uma atualização do aplicativo ... 646
Responder à atualização no dispositivo ... 648
Observar a atualização no aplicativo ... 651
Além da simulação ... 652
Interagir com sombras ... 652
Suporte ao protocolo ... 653
Solicitar e declarar o estado ... 653
Atualizar uma shadow ... 653
Recuperar um documento de shadow ... 656
Excluir dados de sombra ... 657
API REST do Device Shadow ... 659
GetThingShadow ... 660
UpdateThingShadow ... 661
DeleteThingShadow ... 661
ListNamedShadowsForThing ... 662
Tópicos MQTT do Device Shadow ... 663
/get ... 664
/get/accepted ... 665
/get/rejected ... 665
/update ... 666
/update/delta ... 667
/update/accepted ... 668

/update/documents ... 668
/update/rejected ... 669
/delete ... 669
/delete/accepted ... 670
/delete/rejected ... 671
Documentos do serviço Device Shadow ... 671
Exemplos de documentos de sombra ... 672
Propriedades do documento ... 676
Estado delta ... 677
Versionamento de documentos de sombra ... 678
Tokens de cliente em documentos de sombra ... 678
Propriedades vazias do documento de sombra ... 678
Valores de matriz em documentos de sombra ... 679
Mensagens de erro do Device Shadow ... 680
Trabalhos ... 681
Como acessar oAWS IoTempregos ... 681
AWS IoTRegiões de trabalhos e endpoints ... 681
O que é oAWS IoTTrabalhos do? ... 681
Principais conceitos sobre trabalhos ... 682
Estados de execução de trabalhos e trabalhos ... 684
Gerenciar trabalhos ... 688
Assinatura de código para trabalhos ... 688
documento de Job ... 688
Pre-signed URLs ... 688
Criar e gerenciar trabalhos usando o console do ... 690
Criar e gerenciar trabalhos usando a CLI ... 691
Templates de trabalho ... 699
Personalizado eAWSModelos gerenciados ... 699
Usar oAWSModelos gerenciados ... 700
Criar modelos de trabalho personalizados ... 712
Trabalhoconfigurações ... 718
Como as configurações de trabalho funcionam ... 718
Especificar configurações adicionais ... 723
Dispositivos e trabalhos ... 728
Programar dispositivos para funcionarem com trabalhos ... 730
Fluxo de trabalho do dispositivo ... 731
Fluxo de trabalho ... 732
Notificações de trabalhos ... 735
AWS IoTAPIs de trabalhos ... 740
API e tipos de dados de gerenciamento e controle de trabalhos ... 742
APIs e tipos de dados MQTT e HTTPS de dispositivos de trabalho ... 755
Protegendo usuários e dispositivos para trabalhos ... 765
Tipo de política do necessárioAWS IoTTrabalhos do ... 765
Autorizando usuários de Jobs e serviços em nuvem ... 766
Autorizar dispositivos a usar trabalhos ... 772
Limites de trabalhos ... 775
Limites de trabalho ativos e simultâneos ... 775
Encapsulamento seguro do AWS IoT ... 777
O que é tunelamento seguro? ... 777
Conceitos de encapsulamento seguro ... 777
Como o encapsulamento seguro do funciona ... 778
Ciclo de vida do túnel seguro ... 779
AWS IoTTutoriais de encapsulamento seguro ... 779
Abra um túnel e inicie a sessão SSH no dispositivo remoto ... 780
Proxy local ... 782
Como usar o proxy local ... 782
Configurar proxy local para dispositivos que usam proxy da web ... 786

Multiplex fluxos de dados em um túnel seguro ... 791
Snippet de agente de IoT ... 795
Controle de acesso aos túneis ... 796
Pré-requisitos de acesso ao túnel ... 796
Políticas de acesso ao túnel ... 796
Configurar um dispositivo remoto ... 801
Resolvendo problemas de conectividade de tunelamento seguro ... 802
Erro de token de acesso de cliente inválido ... 802
Erro de incompatibilidade de token de cliente ... 802
Problemas de conectividade de dispositivo remoto ... 803
Provisionamento de dispositivos ... 806
Dispositivos de provisionamento emAWS IoT ... 807
APIs de provisionamento de frota ... 807
Provisionar dispositivos que não têm certificados de dispositivo usando o provisionamento de frotas ... 808
Provisionamento por reivindicação ... 808
Provisionamento por usuário confiável ... 810
Usar ganchos de pré-provisionamento com a CLI da AWS ... 812
Provisionamento de dispositivos com certificados de dispositivo ... 814
Provisionamento de uma única coisa ... 815
Provisionamento just-in-time ... 815
Registro em massa ... 818
Modelos de provisionamento ... 819
Seção de parâmetros ... 819
Seção de recursos ... 819
Exemplo de modelo para JITP e registro em massa ... 823
Provisionamento de frota ... 825
Ganchos de pré-provisionamento ... 827
Entrada de gancho de pré-provisão ... 827
Valor de retorno do gancho de pré-provisão ... 828
Exemplo do Lambda do gancho de pré-provisionamento ... 828
API MQTT de provisionamento de dispositivos ... 830
CreateCertificateFromCsr ... 830
CreateKeysAndCertificate ... 832
RegisterThing ... 833
Indexação de frota ... 836
Gerenciamento de indexação de frotas ... 837
Indexação de coisas ... 837
Indexação de grupos de coisas ... 837
Campos gerenciados ... 838
Campos personalizados ... 839
Gerenciar a indexação de coisas ... 840
Gerenciar a indexação de grupos de coisas ... 849
Consulta de dados agregados ... 851
GetStatistics ... 851
GetCardinality ... 853
GetPercentiles ... 854
GetBucketsAggregation ... 856
Autorização ... 856
Sintaxe de consulta ... 857
Recursos compatíveis ... 857
Recursos do não compatíveis ... 857
Observações ... 857
Exemplo de consultas de coisas ... 858
Exemplo de consultas de grupos de coisas ... 861
Métricas de frota ... 862
Tutorial de conceitos básicos ... 862
Gerenciamento de métricas de frota ... 867

Entrega de arquivos baseada em MQTT 872
O que é um fluxo? 872
Gerenciando um fluxo noAWSNuvem 873
Conceder permissões para seus dispositivos 873
Connect os dispositivos aoAWS IoT 874
O uso doAWS IoTEntrega de arquivos baseada em MQTT em dispositivos 874
Use DescribeStream para obter dados de fluxo 874
Obter blocos de dados de um arquivo de fluxo 876
Lidar com erros deAWS IoTEntrega de arquivos baseada em MQTT 880
Um exemplo de caso de uso no FreeRTOS OTA 881
AWS IoT Device Defender 882
Treinamento e certificação da AWS 882
Conceitos básicos do AWS IoT Device Defender 882
Configuração 882
Guia de auditoria 884
Guia de detecção de ML 896
Personalize quando e como você visualizaAWS IoT Device DefenderAuditoria de resultados 919
Auditoria 930
Gravidade do problema 930
Próximas etapas 931
Verificações da auditoria 931
Comandos de auditoria 957
Supressões de descoberta de auditoria 984
Detectar 993
Monitorar o comportamento de dispositivos não registrados 995
Casos de uso de segurança 995
Conceitos 1000
Comportamentos 1001
Detecção de ML 1003
Métricas personalizadas do 1008
Métricas do lado do dispositivo 1013
Métricas do lado da nuvem 1028
Escopo de métricas em perfis de segurança usando dimensões 1035
Permissões 1041
Comandos de detecção 1042
Como usar o AWS IoT Device Defender Detect 1044
Ações de mitigação 1045
Ações de mitigação de auditoria 1046
Detectar ações de mitigação 1048
Como definir e gerenciar ações de mitigação 1048
Aplicar ações de mitigação 1053
Permissões 1058
Comandos de ação de mitigação 1061
Usar o AWS IoT Device Defender com outros serviços da AWS 1062
O uso doAWS IoT Device Defendercom dispositivos em execuçãoAWS IoT Greengrass 1062
O uso doAWS IoT Device Defendercom FreeRTOS e dispositivos incorporados 1062
Uso do AWS IoT Device Defender com a AWS IoT Device Management 1063
Prevenção contra o ataque “Confused deputy” em todos os serviços 1063
Práticas recomendadas de segurança para agentes de dispositivo 1064
Device Advisor 1067
Configuração 1067
Criar uma coisa da IoT 1068
Crie uma função do IAM para ser usada como função de dispositivo 1068
Criar uma política gerenciada personalizada para um usuário do IAM usar o Device Advisor 1069
Criar um usuário do IAM para usar o Device Advisor 1070
Configurar o dispositivo 1070
Primeiros passos com o Device Advisor no console 1071

Fluxo de trabalho do Device 1077
Pré-requisitos 1077
Criar uma definição de suíte de testes 1077
Obtenha uma definição de suíte de testes 1079
Obter um endpoint de teste 1079
Iniciar uma execução de suíte de testes 1080
Faça uma execução de suíte de testes 1080
Pare a execução de um conjunto de testes 1080
Obtenha um relatório de qualificação para uma execução bem-sucedida do conjunto de testes de qualificação 1081
Fluxo de trabalho detalhado do console Device 1081
Pré-requisitos 1082
Criar uma definição de suíte de testes 1082
Iniciar uma execução de suíte de testes 1086
Interromper a execução da suíte de testes (opcional) 1088
Exibir detalhes e registros de execução da suíte de testes 1089
Baixar umAWS IoTRelatório de qualificação do relatório 1090
Casos de testes do Device 1091
Casos de teste do Device Advisor para se qualificar para oAWSPrograma de qualificação do dispositivo. 1091
TLS 1091
MQTT 1095
Shadow 1103
Execução de Job 1104
Políticas e permissões 1105
Mensagens de eventos 1107
Como as mensagens de eventos são geradas 1107
Política para receber mensagens de eventos 1107
Habilite eventos para oAWS IoT 1107
Eventos de registro 1110
Eventos de coisas 1111
Eventos do tipo coisa 1112
Eventos de grupo de coisas 1114
Eventos de trabalho 1118
Eventos de ciclo de vida 1121
Eventos de conexão/desconexão 1121
Eventos de assinatura/cancelamento de assinatura 1124
AWS IoT Corepara LoRaWAN 1126
Introdução 1126
Como usar oAWS IoT Corepara LoRaWAN 1126
AWS IoT CorePara Regiões e endpoints LoRawan 1127
AWS IoT Corepara LoRaWAN 1127
O que é oAWS IoT Corepara LoRaWAN? 1127
O que é LoRaWAN? 1128
ComoAWS IoT Corepara LoRaWAN 1129
Conectando gateways e dispositivos aAWS IoT Core for LoRaWAN 1130
Convenções de nomenclatura para seus dispositivos, gateways, perfis e destinos 1130
Mapeamento de dados do dispositivo para dados de serviço 1130
Usando o console para integrar seu dispositivo e gateway paraAWS IoT Core for LoRaWAN 1131
Descreva suasAWS IoT Corepara recursos LoRaWAN 1131
Incorporar seus gateways paraAWS IoT Core for LoRaWAN 1133
Incorporar seus dispositivosAWS IoT Core for LoRaWAN 1139
Conexão aoAWS IoT Core for LoRaWANPor meio de um endpoint de interface VPC 1150
IntegraçãoAWS IoT Core for LoRaWANEndpoint de API do plano de controle 1151
IntegraçãoAWS IoT Core for LoRaWANEndpoints de API do plano de dados 1154
Como gerenciar gateways comAWS IoT Corepara LoRaWAN 1159
Requisito de software LoRa Basics Station 1160

Usando gateways qualificados a partir doAWSCatálogo de dispositivos para parceiros 1160
Usando protocolos CUPS e LNS 1160
Configure as subbandas e os recursos de filtragem do gateway 1160
Atualize o firmware do gateway usando o serviço CUPS comAWS IoT Core for LoRaWAN 1162
Gerenciamento de dispositivos com o AWS IoT Core for LoRaWAN 1173
Considerações sobre o dispositivo 1173
Usando dispositivos com gateways qualificados paraAWS IoT Core for LoRaWAN 1173
Versão LoRaWAN 1173
Modos de ativação 1173
Classes de dispositivos 1174
Gerencie a comunicação entre seus dispositivos LoraWAN eAWS IoT 1174
Crie grupos multicast para enviar uma carga de downlink para vários dispositivos 1180
Atualizações de firmware over-the-air (FUOTA) paraAWS IoT Core for LoRaWANdispositivos 1191
Monitorando sua frota de recursos sem fio em tempo real usando o analisador de rede 1201
Adicione a função do IAM necessária para o analisador de rede 1202
Crie uma configuração de analisador de rede e adicione recursos 1204
Transmita mensagens de rastreamento do analisador de rede com WebSockets 1209
Visualize e monitore registros de mensagens de rastreamento do analisador de rede em tempo real 1215
Segurança de dados com oAWS IoT Corepara LoRaWAN 1217
Como os dados são protegidos em todo o sistema 1217
Segurança de transporte de dispositivos e gateway LoRaWAN 1218
Integração com o Amazon Sidewalk paraAWS IoT Core 1219
Como integrar seu dispositivo Sidewalk 1219
Dispositivos Sidewalk integrados com Amazon Sidewalk Integration paraAWS IoT Core 1219
Adicione as credenciais da sua conta do Sidewalk 1220
Adicionar um destino para o seu dispositivo Sidewalk 1221
Criar regras para processar mensagens do dispositivo Sidewalk 1223
Connect seu dispositivo Sidewalk e visualize o formato de metadados de uplink 1224
Connect o dispositivo Sidewalk 1224
Exibir formato das mensagens de uplink 1225
Monitorar e registrar em log paraAWS IoT WirelessUsar o Amazon CloudWatch 1226
Configurar o registro em log para o AWS IoT Wireless 1227
Criar uma política e uma função de registro em log paraAWS IoT Wireless 1227
Configurar o registro em log para oAWS IoT Wirelessrecursos 1229
MonitorAWS IoT Wirelessusando CloudWatch Logs 1237
Exibir CloudWatch AWS IoT WirelessEntradas de log do do 1238
Usar o CloudWatch Insights para filtrar logs paraAWS IoT Wireless 1243
Notificações de eventos paraAWS IoT Wireless 1246
Como seus recursos podem ser notificados sobre eventos 1246
Eventos e tipos de recursos 1246
Política para receber notificações de eventos sem fio 1246
Formato de tópicos MQTT para eventos sem fio 1247
Preços para eventos sem fio 1249
Ativar eventos para recursos sem fio 1249
Configurações de evento 1249
Pré-requisitos 1250
Ativar notificações usando oAWS Management Console 1250
Ativar notificações usando oAWS CLI 1251
Notificações de eventos para recursos LoraWAN 1252
Tipos de eventos para recursos da LoRaWAN 1252
LoRaWAN junte-se a eventos 1253
Eventos de status de conexão 1255
Notificações de eventos para recursos Sidewalk 1257
Tipos de eventos para recursos de calçada 1257
Eventos de estado de registro de dispositivos 1257
Eventos de proximidade 1259

Integração do Alexa Voice Service (AVS) aoAWS IoT 1262
Conceitos básicos do Alexa Voice Service (AVS) Integration aoAWS IoTem um dispositivo NXP 1263
Visualizar a integração do Alexa Voice Service (AVS) aoAWS IoTCom uma conta do NXP pré-configurada 1264
Interaja com a Alexa 1275
Use a suaAWSe contas de desenvolvedor do Alexa Voice Service para configurar o AVS aoAWS IoT 1277
AWS IoTSDKs de dispositivos, SDKs móveis eAWS IoTCliente de dispositivo 1278
SDKs de dispositivo da AWS IoT 1278
SDK de dispositivo da AWS IoT para C incorporado 1279
AnteriormenteAWS IoTVersões dos SDKs do 1280
SDKs móveis do AWS 1280
AWS IoTCliente de dispositivo 1281
Solução de problemas 1282
Diagnóstico de problemas de conectividade 1282
Conexão 1282
Autenticação 1283
Autorização 1284
Segurança e identidade 1284
Diagnosticar problemas de regras 1285
Configurar o CloudWatch Logs para solução de problemas 1285
Diagnóstico de serviços externos 1286
Diagnóstico de problemas de SQL 1286
Diagnosticar problemas com shadows 1286
Diagnosticar problemas de ação da Salesforce 1287
Rastreamento da execução 1287
Sucesso e falha da ação 1288
Guia de solução de problemas de indexação de frota 1288
Solução de problemas de consultas de agregação para o serviço de indexação de frota 1288
Solução de problemas de métricas de frota 1289
Solução de problemas “Limite de fluxo excedido para oAWSconta” 1290
Guia de solução de problemas do AWS IoT Device Defender 1290
AWS IoTGuia de solução de problemas do Device Advisor 1294
Solução de problemas de desconexões de frota de dispositivos 1295
Erros do AWS IoT 1296
AWS IoTCotas do 1298
..... mccxcix

O que é o AWS IoT?

A AWS IoT fornece os serviços de nuvem que conectam seus dispositivos IoT a outros dispositivos e aos serviços de nuvem da AWS. A AWS IoT fornece software para dispositivos que pode ajudar você a integrar seus dispositivos IoT a soluções baseadas em AWS IoT. Se seus dispositivos puderem se conectar ao AWS IoT, o AWS IoT poderá conectá-los aos serviços em nuvem que a AWS fornece.

Para uma introdução prática aoAWS IoT, visiteConceitos básicos do AWS IoT Core (p. 18).

AWS IoTpermite que você selecione o mais apropriado e up-to-date tecnologias para sua solução. Para ajudá-lo a gerenciar e dar suporte a seus dispositivos IoT em campo,AWS IoT Coresuporta estes protocolos:

- MQTT (Enfileiramento de Mensagens e Transporte de Telemetria) (p. 86)
- MQTT sobre WSS (Websockets seguros) (p. 86)
- HTTPS (Hypertext Transfer Protocol - Seguro) (p. 97)
- LoRaWAN (Rede de Área Ampla de Longo Alcance) (p. 1126)

OAWS IoT CoreO agente de mensagens oferece suporte a dispositivos e clientes que usam protocolos MQTT e MQTT sobre WSS para publicar e assinar mensagens. Ele também oferece suporte a dispositivos e clientes que usam o protocolo HTTPS para publicar mensagens.

AWS IoT Corepara LoRaWAN ajuda você a conectar e gerenciar dispositivos sem fio LoRaWAN (rede de área larga de longo alcance de baixa potência).AWS IoT Corepara LorAwan substitui a necessidade de você desenvolver e operar um LoraWAN Network Server (LNS).

Se você não precisarAWS IoTrecursos como comunicações de dispositivos,regras (p. 478), ouempregos (p. 681)consulteAWSMensagenspara obter informações sobre outrosAWS IoTserviços de mensagens que podem se adequar melhor às suas necessidades.

Como seus dispositivos e aplicativos acessamAWS IoT

AWS IoTFornece as seguintes interfaces paraTutoriais do AWS IoT (p. 125):

- AWS IoTSDKs de dispositivo da— Crie aplicativos em seus dispositivos que enviam e recebem mensagens deAWS IoT. Para obter mais informações, consulteAWS IoTSDKs de dispositivos, SDKs móveis eAWS IoTCliente de dispositivo (p. 1278)
- AWS IoT Corepara LoRaWAN—Conecte e gerencie seus dispositivos e gateways WAN de longo alcance (LoRaWAN) usandoAWS IoT Corepara LoRaWAN (p. 1126).
- AWS Command Line Interface (AWS CLI) — executa comandos para a AWS IoT no Windows, no macOS e no Linux. Esses comandos permitem que você crie e gerencie objetos, certificados, regras, tarefas e políticas. Para começar a usar, consulte o Guia do usuário do AWS Command Line Interface. Para obter mais informações sobre os comandos doAWS IoTconsulteiotnoAWS CLIReferência de comando.
- API da AWS IoT— Cria seus aplicativos para IoT usando solicitações HTTP ou HTTPS. Essas ações da API permitem que você crie e gerencie de modo programático objetos, certificados, regras e políticas. Para obter mais informações sobre as ações de API para AWS IoT, consulte Ações na Referência de API para AWS IoT.
- AWSSDKs da— Crie aplicativos para a IoT usando APIs específicas de uma linguagem. Esses SDKs encapsulam a API HTTP/HTTPS e permitem que você programe em qualquer uma das linguagens suportadas. Para obter mais informações, consulte AWSSDKs e ferramentas da.

Você também pode acessarAWS IoTpor meio doAWS IoTconsole, que fornece uma interface gráfica do usuário (GUI) através da qual você pode configurar e gerenciar objetos, certificados, regras, trabalhos, políticas e outros elementos de suas soluções de IoT.

O queAWS IoTpode fazer

Este tópico descreve algumas das soluções que você pode precisar doAWS IoToferece suporte a.

IoT na indústria

Estes são alguns exemplos deAWS IoTSoluções paraCasos de uso industrialque aplicam tecnologias de IoT para melhorar o desempenho e a produtividade dos processos industriais.

Soluções para casos de uso industrial

- Usar oAWS IoTpara construir modelos de qualidade preditiva em operações industriais

  Veja comoAWS IoTpode coletar e analisar dados de operações industriais para construir modelos de qualidade preditiva. Saiba mais
- Usar oAWS IoTpara apoiar a manutenção preditiva em operações industriais

  Veja comoAWS IoTpode ajudar a planejar manutenção preventiva para reduzir o tempo de inatividade não planejado. Saiba mais

IoT na automação residencial

Estes são alguns exemplos deAWS IoTSoluções paraCasos de uso de automação residencialque aplicam tecnologias de IoT para criar aplicativos de IoT escaláveis que automatizam atividades domésticas usando dispositivos domésticos conectados.

Soluções para automação residencial

- Usar oAWS IoTem sua casa conectada

  Veja comoAWS IoTpode fornecer soluções integradas de automação residencial.
- Usar oAWS IoTpara fornecer segurança doméstica e monitoramento

  Veja comoAWS IoTpode aplicar aprendizado de máquina e computação de borda à sua solução de automação residencial.

Para obter uma lista de soluções para casos de uso industrial, de consumo e comercial, consulte oAWS IoTrepositório de soluções.

Como o AWS IoT funciona

AWS IoTfornece serviços em nuvem e suporte a dispositivos que você pode usar para implementar soluções de IoT.AWSfornece muitos serviços em nuvem para oferecer suporte a aplicativos baseados em IoT. Então, para ajudá-lo a entender por onde começar, esta seção fornece um diagrama e uma definição de conceitos essenciais para apresentá-lo ao universo da IoT.

O universo do IoT

Em geral, a Internet das Coisas (IoT) consiste nos principais componentes mostrados neste diagrama.

Apps

Os aplicativos dão aos usuários finais acesso aos dispositivos IoT e aos recursos fornecidos pelos serviços de nuvem aos quais esses dispositivos estão conectados.

Serviços de nuvem

Os serviços em nuvem são serviços distribuídos de armazenamento e processamento de dados em larga escala conectados à Internet. Os exemplos incluem:

- Serviços de gerenciamento e conexão IoT.

  AWS IoTé um exemplo de um serviço de gerenciamento e conexão IoT.
- Serviços de computação, como o Amazon Elastic Compute Cloud eAWS Lambda.
- Serviços de banco de dados, como o Amazon DynamoDB

Comunicações

Os dispositivos se comunicam com serviços em nuvem usando várias tecnologias e protocolos. Os exemplos incluem:

- Internet Wi-Fi/banda larga
- Dados celulares de banda larga
- Dados celulares de banda estreita
- Rede de área ampla de longo alcance (LOraWAN)
- Comunicações RF exclusivas

Dispositivos

Um dispositivo é um tipo de hardware que gerencia interfaces e comunicações. Os dispositivos geralmente estão localizados próximos às interfaces do mundo real que monitoram e controlam. Os dispositivos podem incluir recursos de computação e armazenamento, como microcontroladores, CPU, memória. Os exemplos incluem:

- Raspberry PI
- Arduino
- Assistentes de interface de voz
- LoRaWAN e dispositivos
- Dispositivos Amazon Sidewalk
- Dispositivos de IoT personalizados

Interfaces

Uma interface é um componente que conecta um dispositivo ao mundo físico.

- Interfaces do usuário

  Componentes que permitem que dispositivos e usuários se comuniquem entre si.

  - Interfaces de entrada

    Permitir que um usuário se comunique com um dispositivo

    Exemplos: teclado, botão
  - Interfaces de saída

    Permitir que um dispositivo se comunique com um usuário

    Exemplos: Visor alfa-numérico, exibição gráfica, luz indicadora, alarme
- Sensores

  Inserir componentes que medem ou sentem algo no mundo exterior de uma maneira que um dispositivo entenda. Os exemplos incluem:

  - Sensor de temperatura (converte a temperatura em um sinal analógico ou digital)
  - Sensor de umidade (converte a umidade relativa em um sinal analógico ou digital)
  - Conversor analógico para digital (converte uma tensão analógica em um valor numérico)
  - Unidade de medição de distância ultra-sônica (converte uma distância em um valor numérico)
  - Sensor óptico (converte um nível de luz em um valor numérico)
  - Câmera (converte dados de imagem em dados digitais)
- Atuadores

  Componentes de saída que o dispositivo pode usar para controlar algo no mundo exterior. Os exemplos incluem:

  - Motores de passo (converta sinais elétricos em movimento)
  - Relés (controle de altas tensões e correntes elétricas)

AWS IoTVisão geral dos serviços

No universo da IoT,AWS IoTfornece os serviços que suportam os dispositivos que interagem com o mundo e os dados que passam entre eles eAWS IoT.AWS IoTé composto pelos serviços mostrados nesta ilustração para dar suporte à sua solução de IoT.

AWS IoTsoftware de dispositivo

AWS IoTfornece este software para suportar seus dispositivos IoT.

SDKs de dispositivo da AWS IoT

OAWS IoTSDKs de dispositivos e dispositivos móveis (p. 1278)ajuda você a conectar seus dispositivos com eficiênciaAWS IoT. Os SDKs de dispositivos AWS IoT e dispositivos móveis incluem bibliotecas de código aberto, guias de desenvolvedor com exemplos e guias de portabilidade para que você possa criar produtos ou soluções inovadoras da IoT nas plataformas de hardware de sua preferência.

AWS IoT Device Tester

AWS IoT Device Testerpara FreeRTOS eAWS IoT Greengrassé uma ferramenta de automação de testes para microcontroladores.AWS IoT Device Testertesta seu dispositivo para determinar se ele executará o FreeRTOS ouAWS IoT Greengrasse interoperar comAWS IoTServiços da .

AWS IoTExpressLink

AWS IoT ExpressLink alimenta uma variedade de módulos de hardware desenvolvidos e oferecidos pelaAWSParceiros. Os módulos de conectividade incluemAWS-software validado, tornando mais rápido e fácil para você conectar dispositivos com segurança à nuvem e integrar-se perfeitamente a uma variedade deAWSServiços da . Para obter mais informações, acesse aAWS IoTExpressLinkpágina de visão geral ou veja aAWS IoT ExpressLink Guia do programador do.

AWS IoTGreengrass

AWS IoTGreengrassestendeAWS IoTPara dispositivos de borda para que eles possam atuar localmente sobre os dados que geram e usam a nuvem para gerenciamento, análise e armazenamento resiliente. comAWS IoTGreengrass, dispositivos conectados podem ser

executadosAWS LambdaAs funções do Docker ou ambos, executam previsões baseadas em modelos de machine learning, mantêm o sincronismo de dados de dispositivos e comunicam-se com outros dispositivos de forma segura, mesmo quando não estiverem conectados à Internet.

FreeRTOS

FreeRTOSé um sistema operacional de código aberto em tempo real para microcontroladores que permite incluir dispositivos de borda pequenos e de baixo consumo em sua solução de IoT. O FreeRTOS inclui um kernel e um conjunto crescente de bibliotecas de software que suportam muitos aplicativos. Os sistemas FreeRTOS podem conectar com segurança seus dispositivos pequenos e de baixa potência aAWS IoTe suporte a dispositivos de borda mais potentes em execuçãoAWS IoT Greengrass.

AWS IoTServiços de controle

Connect ao seguinteAWS IoTserviços para gerenciar os dispositivos em sua solução de IoT.

AWS IoT Core

AWS IoT CoreO é um serviço de nuvem gerenciada que permite que dispositivos conectados interajam com segurança com aplicativos de nuvem e outros dispositivos.AWS IoT Corepode suportar muitos dispositivos e mensagens, e ele pode processar e encaminhar essas mensagens paraAWS IoTendpoints e outros dispositivos. comAWS IoT Core, seus aplicativos podem interagir com todos os seus dispositivos mesmo quando eles não estão conectados.

AWS IoT CoreDevice Advisor

AWS IoT CoreDevice Advisoré um recurso de teste totalmente gerenciado baseado em nuvem para validar dispositivos IoT durante o desenvolvimento de software de dispositivos. O Device Advisor fornece testes pré-criados que você pode usar para validar dispositivos IoT para conectividade confiável e segura comAWS IoT Core, antes de implantar dispositivos na produção.

AWS IoT Device Defender

AWS IoTDevice Defenderajuda você a proteger sua frota de dispositivos IoT.AWS IoT O Device Defender audita continuamente suas configurações de IoT para garantir que elas não estejam se desviando das melhores práticas de segurança.AWS IoT O Device Defender envia um alerta quando detecta quaisquer lacunas na sua configuração de IoT que possam criar um risco de segurança, como certificados de identidade sendo compartilhados em vários dispositivos ou um dispositivo com um certificado de identidade revogado tentando se conectar aoAWS IoT Core.

AWS IoTGerenciamento de dispositivos

AWS IoTGerenciamento de dispositivosos serviços ajudam você a rastrear, monitorar e gerenciar a infinidade de dispositivos conectados que compõem suas frotas de dispositivos.AWS IoT Os serviços de gerenciamento de dispositivos ajudam você a garantir que seus dispositivos IoT funcionem de forma adequada e segura depois de serem implantados. Eles também fornecem tunelamento seguro para acessar seus dispositivos, monitorar sua integridade, detectar e solucionar problemas remotamente, bem como serviços para gerenciar atualizações de software e firmware do dispositivo.

Serviços de dados do AWS IoT

Analise os dados dos dispositivos em sua solução de IoT e tome as medidas apropriadas usando o seguinteAWS IoTServiços da .

Amazon Kinesis Video Streams

Amazon Kinesis Video Streamspermite transmitir vídeo ao vivo de dispositivos para oAWSNuvem, onde é armazenada, criptografada e indexada de forma duradoura, permitindo que você acesse seus

dados por meio de easy-to-use APIs. Você pode usar o Amazon Kinesis Video Streams para capturar enormes quantidades de dados de vídeo ao vivo de milhões de fontes, incluindo smartphones, câmeras de segurança, webcams, câmeras embutidas em carros, drones e muito mais. O Amazon Kinesis Video Streams permite que você reproduza vídeos para visualização ao vivo e sob demanda, e crie rapidamente aplicativos que aproveitem a visão computacional e a análise de vídeo por meio da integração com o Amazon Rekognition Video e bibliotecas para estruturas de ML. Você também pode enviar dados diferentes de vídeo e não serializados em tempo, como dados de áudio, imagens térmicas, dados de profundidade, dados RADAR e muito mais.

Amazon Kinesis Video Streams com WebRTC

Amazon Kinesis Video Streams com WebRTCfornece uma implementação WebRTC compatível com padrões como um recurso totalmente gerenciado. Você pode usar o Amazon Kinesis Video Streams com WebRTC para transmitir mídia ao vivo com segurança ou realizar interações bidirecionais de áudio ou vídeo entre qualquer dispositivo de IoT de câmera e players móveis ou web compatíveis com WebRTC. Como um recurso totalmente gerenciado, você não precisa criar, operar ou dimensionar nenhuma infraestrutura de nuvem relacionada ao WebRTC, como servidores de sinalização ou retransmissão de mídia para transmitir mídia com segurança entre aplicativos e dispositivos. Usando o Amazon Kinesis Video Streams com WebRTC, você pode facilmente criar aplicativos para o tempo real peer-to-peer streaming de mídia ou interatividade de áudio ou vídeo em tempo real entre dispositivos IoT de câmera, navegadores da Web e dispositivos móveis para uma variedade de casos de uso.

AWS IoT Analytics

AWS IoTAnálisePermite executar e operacionalizar com eficiência análises sofisticadas em volumes massivos de dados de IoT não estruturados.AWS IoT O Analytics automatiza cada etapa difícil necessária para analisar os dados de dispositivos IoT.AWS IoT O Analytics filtra, transforma e enriquece os dados da IoT antes de armazená-los em um datastore de séries temporais para serem analisados. Você pode analisar os dados executando consultas únicas ou agendadas usando o mecanismo de consultas SQL integrado ou o machine learning.

Eventos do AWS IoT

AWS IoTEventosdetecta e responde a eventos de sensores e aplicativos de IoT. Eventos são padrões de dados que identificam circunstâncias mais complicadas do que o esperado, como detectores de movimento usando sinais de movimento para ativar luzes e câmeras de segurança.AWS IoT Eventos monitora continuamente dados de vários sensores e aplicativos de IoT e se integram a outros serviços, comoAWS IoT Core, IoT SiteWise, DynamoDB e outros para permitir a detecção precoce e insights exclusivos.

AWS IoTFleetWise

AWS IoTFleetWiseé um serviço gerenciado que você pode usar para coletar e transferir dados do veículo para a nuvem quase em tempo real. comAWS IoTFleetWise, você pode facilmente coletar e organizar dados de veículos que usam diferentes protocolos e formatos de dados. AWS IoT FleetWise ajuda a transformar mensagens de baixo nível em valores legíveis por humanos e padronizar o formato de dados na nuvem para análises de dados. Você também pode definir esquemas de coleta de dados para controlar quais dados coletar em veículos e quando transferi-los para a nuvem.

AWS IoTSiteWise

AWS IoTSiteWisecoleta, armazena, organiza e monitora dados passados de equipamentos industriais por mensagens MQTT ou APIs em escala, fornecendo software que é executado em um gateway em suas instalações. O gateway se conecta com segurança aos seus servidores de dados locais e automatiza o processo de coleta e organização dos dados e enviá-los para oAWSCloud.

AWS IoT TwinMaker

AWS IoTTwinMakerconstrói gêmeos digitais operacionais de sistemas físicos e digitais. AWS IoT TwinMaker cria visualizações digitais usando medições e análises de uma variedade de sensores, câmeras e aplicativos corporativos do mundo real para ajudá-lo a acompanhar sua fábrica física,

construção ou fábrica industrial. Você pode usar dados do mundo real para monitorar operações, diagnosticar e corrigir erros e otimizar as operações.

Serviços da AWS IoT Core

AWS IoT Corefornece os serviços que conectam seus dispositivos IoT aoAWSNuvem para que outros serviços e aplicativos em nuvem possam interagir com seus dispositivos conectados à Internet.

A próxima seção descreve cada uma dasAWS IoT Coreserviços mostrados na ilustração.

AWS IoT CoreServiços de mensagens

OAWS IoT Coreos serviços de conectividade fornecem comunicação segura com os dispositivos IoT e gerenciam as mensagens que passam entre eles eAWS IoT.

Gateway do dispositivo

Permite que dispositivos se comuniquem com a AWS IoT com segurança e eficiência. A comunicação do dispositivo é protegida por protocolos seguros que usam certificados X.509.

Operador de mensagens

Fornece um mecanismo seguro para dispositivos e aplicativos da AWS IoT a fim de publicar e receber mensagens um do outro. Você pode usar o protocolo MQTT diretamente ou MQTT sobre WebSocket para publicar e assinar. Para obter mais informações sobre os protocolos queAWS IoTsuporta, consultethe section called “Protocolos de comunicação de” (p. 83). Os dispositivos e clientes também podem usar a interface REST HTTP para publicar dados no agente de mensagens.

O corretor de mensagens distribui dados do dispositivo para dispositivos que se inscreveram nele e para outrosAWS IoT Coreserviços, como o serviço Device Shadow e o mecanismo de regras.

AWS IoT Corepara LoRaWAN

AWS IoT Corepara LoraWAN possibilita a configuração de uma rede LoRaWAN privada conectando seus dispositivos e gateways LoraWAN aAWSsem a necessidade de desenvolver e operar um LoraWAN Network Server (LNS). As mensagens recebidas dos dispositivos LoraWAN são enviadas para o mecanismo de regras, onde podem ser formatadas e enviadas para outrosAWS IoTServiços da .

Mecanismo de regras

O mecanismo de regras conecta dados do corretor de mensagens a outrosAWS IoTserviços para armazenamento e processamento adicional. Por exemplo, você pode inserir, atualizar ou consultar uma tabela do DynamoDB ou chamar uma função do Lambda baseada em uma expressão definida no mecanismo de regras. Você pode usar uma linguagem baseada em SQL para selecionar dados de cargas de mensagem, processar e enviar os dados para outros serviços, como Amazon Simple Storage Service (Amazon S3), Amazon DynamoDB eAWS Lambda. Você também pode criar regras que republicam mensagens para o corretor de mensagens e para outros assinantes. Para obter mais informações, consulteRegras para AWS IoT (p. 478)

AWS IoT CoreServiços de controle

OAWS IoT Coreos serviços de controle fornecem recursos de segurança, gerenciamento e registro do dispositivo.

Serviço de autenticação personalizada

Você pode definir autorizadores personalizados que permitem que você gerencie sua própria estratégia de autenticação e autorização usando um serviço de autenticação personalizada e uma função do Lambda. Os autorizadores personalizados permitem que a AWS IoT autentique seus dispositivos e autorize operações usando as estratégias de autenticação e autorização do token do portador.

Autorizadores personalizados podem implementar várias estratégias de autenticação; por exemplo, verificação JSON Web Token ou callout do provedor OAuth. Eles devem retornar documentos de política usados pelo gateway de dispositivos para autorizar operações MQTT. Para obter mais informações, consulteAutenticação personalizada (p. 319)

Serviço de provisionamento de dispositivos

Permite provisionar dispositivos usando um modelo que descreve os recursos necessários para o dispositivo: umobjeto de coisa, um certificado e uma ou mais políticas. Um objeto da coisa é uma entrada no registro que contém atributos que descrevem um dispositivo. Os dispositivos usam certificados para autenticação com a AWS IoT. As políticas determinam quais operações um dispositivo pode executar na AWS IoT.

Os modelos contêm variáveis que são substituídas por valores em um dicionário (mapa). Você pode usar o mesmo modelo para provisionar vários dispositivos bastando passar diferentes valores para as variáveis do modelo no dicionário. Para obter mais informações, consulteProvisionamento de dispositivos (p. 806)

Registro de grupo

Os grupos permitem gerenciar vários dispositivos simultaneamente, categorizando-os em grupos. Os grupos também podem conter grupos — você pode criar uma hierarquia de grupos. Qualquer ação executada em um grupo pai será aplicada a seus grupos filho. A mesma ação também se aplica a todos os dispositivos no grupo pai e a todos os dispositivos nos grupos filhos. As permissões concedidas a um grupo serão aplicadas a todos os dispositivos no grupo e a todos os seus grupos filho. Para obter mais informações, consulteGerenciamento de dispositivos com o AWS IoT (p. 265)

Serviço Jobs

Permite definir um conjunto de operações remotas que são enviadas e executadas em um ou mais dispositivos conectados aoAWS IoT. Por exemplo, você pode definir um trabalho que instrui um conjunto de dispositivos a baixar e instalar um aplicativo ou atualizações de firmware, reinicializar, alternar certificados ou executar operações de solução de problemas remotamente.

Para criar um trabalho, você especifica uma descrição das operações remotas a serem executadas e uma lista de destinos que devem executá-las. Os destinos podem ser dispositivos individuais, grupos ou ambos. Para obter mais informações, consulteTrabalhos (p. 681)

Registro

Organiza os recursos associados a cada dispositivo naAWSCloud. É possível registrar seus dispositivos e associar até três atributos personalizados a cada um. Você também pode associar certificados e IDs de cliente MQTT a cada dispositivo para melhorar sua capacidade de gerenciá-los e de solucionar seus problemas. Para obter mais informações, consulteGerenciamento de dispositivos com o AWS IoT (p. 265)

Serviço de segurança e identidade

Fornece responsabilidade compartilhada para segurança noAWSCloud. Seus dispositivos devem manter as credenciais seguras para enviar dados com segurança ao operador de mensagens. O corretor de mensagens e o mecanismo de regras usamAWSRecursos de segurança para enviar dados com segurança para dispositivos ou outrosAWSServiços da . Para obter mais informações, consulteAutenticação (p. 294)

Serviços de dados do AWS IoT Core

OAWS IoT Coreos serviços de dados ajudam suas soluções de IoT a fornecer uma experiência de aplicativo confiável, mesmo com dispositivos que nem sempre estão conectados.

Device Shadow

Um documento JSON usado para armazenar e recuperar as informações do estado atual de um dispositivo.

Serviço Device Shadow

O serviço Device Shadow mantém o estado de um dispositivo para que os aplicativos possam se comunicar com um dispositivo, independentemente de o dispositivo estar online ou não. Quando um dispositivo está offline, o serviço Device Shadow gerencia seus dados para aplicativos conectados. Quando o dispositivo se reconecta, ele sincroniza seu estado com o de sua sombra no serviço Device Shadow. Seus dispositivos também podem publicar o estado atual deles em uma sombra para serem usados por aplicativos ou outros dispositivos que talvez não estejam conectados o tempo todo. Para obter mais informações, consulteServiço Device Shadow do AWS IoT (p. 633)

AWS IoT Coreserviço de suporte

Integração do Alexa Voice Service (AVS) paraAWS IoT

Traz o Alexa Voice para qualquer dispositivo conectado. AVS paraAWS IoTReduz o custo e a complexidade da integração da Alexa. Esse recurso usaAWS IoTPara descarregar tarefas de áudio de memória e computação intensivas do dispositivo para a nuvem. Devido à redução resultante no custo da lista de materiais de engenharia (EBOM), os fabricantes de dispositivos podem trazer a Alexa para dispositivos de IoT com restrições de recursos e permitir que os consumidores falem diretamente com a Alexa em cômodos da casa, no escritório ou em quartos de hotel para uma experiência ambiente.

AVS paraAWS IoTO permite a funcionalidade incorporada do Alexa em MCUs, como a classe ARM Cortex M com menos de 1 MB de RAM incorporada. Para fazer isso, o AVS descarrega na nuvem tarefas de memória e computação para um dispositivo virtual com Alexa integrada. Isso reduz o custo da EBOM em até 50%. Para obter mais informações, consulteIntegração do Alexa Voice Service (AVS) aoAWS IoT (p. 1262)

Integração com o Amazon Sidewalk paraAWS IoT Core

Amazon Sidewalké uma rede compartilhada que melhora as opções de conectividade para ajudar os dispositivos a trabalhar melhor juntos. O Amazon Sidewalk oferece suporte a uma ampla variedade de dispositivos do cliente, como aqueles que localizam animais de estimação ou objetos de valor, aqueles

que fornecem segurança doméstica inteligente e controle de iluminação e aqueles que fornecem diagnósticos remotos para aparelhos e ferramentas. Integração com o Amazon Sidewalk paraAWS IoT Corepossibilita que os fabricantes de dispositivos adicionem sua frota de dispositivos Sidewalk aoAWS IoTCloud.

Para obter mais informações, consulteIntegração com o Amazon Sidewalk paraAWS IoT Core (p. 1219)

Saiba mais sobre o AWS IoT

Este tópico ajuda você a se familiarizar com o mundo doAWS IoT. Você pode obter informações gerais sobre como as soluções de IoT são aplicadas em vários casos de uso, recursos de treinamento, links para mídias sociais paraAWS IoTe todos os outrosAWSserviços e uma lista de serviços e protocolos de comunicação queAWS IoTUso do.

Recursos de treinamento paraAWS IoT

Fornecemos esses cursos de treinamento para ajudá-lo a saber mais sobreAWS IoTE como aplicá-los ao design da sua solução.

- Introdução à AWS IoT

  Visão geral em vídeo doAWS IoTe seus principais serviços.
- Mergulhe profundamente emAWS IoTAutenticação e autorização

  Um curso avançado que explora os conceitos deAWS IoTAutorização e autenticação. Você aprenderá a autenticar e autorizar clientes a acessar oAWS IoTAPIs de plano de controle e plano de dados.
- Série da Fundação Internet das Coisas

  Um caminho de aprendizado dos módulos de eLearning de IoT em diferentes tecnologias e recursos de IoT.

AWS IoTrecursos e guias

Estes são recursos técnicos aprofundados sobre aspectos específicos doAWS IoT.

- Lens IoT —AWS IoTEstrutura Well-Architected

  Um documento que descreve as melhores práticas para arquitetar seus aplicativos de IoT emAWS.
- Projetando tópicos do MQTT paraAWS IoT Core

  Um documento PDF que descreve as melhores práticas para projetar tópicos do MQTT emAWS IoT Coree alavancandoAWS IoT Corerecursos com MQTT.
- Fabricação e provisionamento de dispositivos com certificados X.509 emAWS IoT Core

  Um documento PDF que descreve as diferentes maneiras queAWS IoTfornece para fornecer grandes frotas de dispositivos.
- AWS IoT Core Device Advisor

  AWS IoT CoreO Device Advisor fornece testes pré-criados que você pode usar para validar dispositivos IoT para práticas recomendadas de conectividade confiáveis e seguras comAWS IoT Core, antes de implantar dispositivos na produção.
- Recursos da AWS IoT

Recursos específicos da IoT, como guias técnicos, arquiteturas de referência, eBooks e postagens de blog com curadoria, apresentados em um índice pesquisável.

- Atlas de IoT

  Visões gerais sobre como resolver problemas comuns de design de IoT. OAtlas de IoTfornece análises detalhadas sobre os desafios de projeto que você provavelmente encontrará ao desenvolver sua solução de IoT.
- AWSWhitepapers e guias

  Nossa coleção atual de whitepapers e guias sobreAWS IoTe outrosAWSTecnologias.

AWS IoTnas redes sociais

Esses canais de mídia social fornecem informações sobreAWS IoTeAWSTópicos relacionados.

- A Internet das Coisas emAWS IoT— Blog oficial
- AWS IoTVídeos no canal da Amazon Web Services no YouTube

Essas contas de mídia social cobrem todosAWSserviços, incluindoAWS IoT

- O canal da Amazon Web Services no YouTube
- Amazon Web Services no Twitter
- Amazon Web Services no Facebook
- Amazon Web Services no Instagram
- Amazon Web Services no LinkedIn

AWSserviços usados peloAWS IoT CoreMecanismo de regras

OAWS IoT Coremecanismo de regras pode se conectar a estesAWSServiços da .

- Amazon DynamoDB

  O Amazon DynamoDB é um serviço de banco de dados NoSQL escalonável que proporciona um desempenho de banco de dados rápido e previsível.
- Amazon Kinesis

  O Amazon Kinesis facilita a coleta, o processamento e a análise de dados de streaming em tempo real para que você possa obter insights oportunos e reagir rapidamente a novas informações. O Amazon Kinesis pode ingerir dados em tempo real, como vídeo, áudio, registros de aplicativos, fluxos de cliques de sites e dados de telemetria de IoT para aprendizado de máquina, análises e outros aplicativos.
- AWS Lambda

  O AWS Lambda permite que você execute código sem provisionar ou gerenciar servidores. Você pode configurar seu código para ser acionado automaticamente doAWS IoTdados e eventos ou chame-os diretamente de um aplicativo web ou móvel.
- Amazon Simple Storage Service

  O Amazon Simple Storage Service (Amazon S3) pode armazenar e recuperar qualquer volume de dados, a qualquer momento, de qualquer lugar na Web.AWS IoTAs regras podem enviar dados ao Amazon S3 para armazenamento.

- Amazon Simple Notification Service

  O Amazon Simple Notification Service (Amazon SNS) é um serviço web que permite que os aplicativos, os usuários finais e os dispositivos enviem e recebam notificações da nuvem.
- Amazon Simple Queue Service

  O Amazon Simple Queue Service (Amazon SQS) é um serviço de filas de mensagens que desacoplam e dimensionam microsserviços, sistemas distribuídos e aplicativos sem servidor.
- Amazônia OpenSearch Serviço

  Amazônia OpenSearch O Service (OpenSearch Service) é um serviço gerenciado que facilita a implantação, a operação e o dimensionamento do OpenSearch, um popular mecanismo de pesquisa e análise de código aberto.
- Amazon Machine Learning

  O Amazon Machine Learning pode criar modelos de Machine Learning (ML) ao encontrar padrões nos dados da IoT. O serviço usa esses modelos para processar novos dados e gerar previsões para seu aplicativo.
- Amazon CloudWatch

  Amazônia CloudWatch Fornece uma solução de monitoramento confiável, escalonável e flexível para ajudar a configurar, gerenciar e dimensionar sua própria infraestrutura e sistemas de monitoramento.

Protocolos de comunicação suportados porAWS IoT Core

Esses tópicos fornecem informações adicionais sobre os protocolos de comunicação usados peloAWS IoT. Para obter mais informações sobre os protocolos usados peloAWS IoTe conectando dispositivos e serviços aAWS IoTconsulteComo conectar-se ao AWS IoT Core (p. 71).

- MQTT (Transporte de telemetria de enfileiramento de mensagens)

  A página inicial do site MQTT.org onde você pode encontrar as especificações do protocolo MQTT. Para obter mais informações sobre comoAWS IoTsuporta MQTT, consulteMQTT (p. 86).
- HTTPS (Hypertext Transfer Protocol - Seguro)

  Dispositivos e aplicativos podem acessarAWS IoTserviços usando HTTPS.
- LoRaWAN (Rede de Área Ampla de Longo Alcance)

  Dispositivos e gateways LoRaWAN podem se conectar aAWS IoT CoreusandoAWS IoT Corepara LoRaWAN.
- TLS (Transport Layer Security) v1.2

  A especificação do TLS v1.2 (RFC 5246).AWS IoTO usa o TLS v1.2 para estabelecer conexões seguras entre dispositivos eAWS IoT.

Novidades doAWS IoTconsole

Estamos em processo de atualização da interface de usuário doAWS IoTconsole para uma nova experiência. Estamos atualizando a interface do usuário em etapas, então algumas páginas no console terão uma nova experiência, algumas podem ter tanto a experiência original quanto a nova, e algumas podem ter apenas a experiência original.

Esta tabela exibe o estado de áreas individuais doAWS IoTinterface de usuário do console a partir de 27 de janeiro de 2022.

AWS IoTstatus da interface de usuário do console

Página do console	Experiência original	Nova experiência	Comentários
Monitor	Não disponível	Disponível	
Atividades	Não disponível	Disponível	
Integração- Comece a usar	Não disponível	Sem disponibilidade no momento	
Integração- Modelos de provisionamento de frota	Disponível	Sem disponibilidade no momento	
Gerencie- Coisas	Disponível	Disponível	
Gerencie- Tipos	Disponível	Disponível	
Gerencie- Grupos de coisas	Disponível	Disponível	
Gerencie- Grupos de faturamento	Disponível	Disponível	
Gerencie- Empregos	Disponível	Disponível	
Gerencie- Templates de Job	Não disponível	Disponível	
Gerencie- Túneis	Não disponível	Disponível	
Hub da frota- Comece a usar	Não disponível	Disponível	Não disponível em todasRegiões da AWS
Hub da frota- Aplicações	Não disponível	Disponível	Não disponível em todasRegiões da AWS
Greengrass- Conceitos básicos	Não disponível	Disponível	Não disponível em todasRegiões da AWS
Greengrass- Dispositivos principais do	Não disponível	Disponível	Não disponível em todasRegiões da AWS
Greengrass- Componentes	Não disponível	Disponível	Não disponível em todasRegiões da AWS
Greengrass- Implantações	Não disponível	Disponível	Não disponível em todasRegiões da AWS
Greengrass- Clássico (V1)	Disponível	Não disponível	Não disponível em todasRegiões da AWS
Conectividade sem fio- Introdução	Não disponível	Disponível	Não disponível em todasRegiões da AWS
Conectividade sem fio- Gateways	Não disponível	Disponível	Não disponível em todasRegiões da AWS

Página do console	Experiência original	Nova experiência	Comentários
Conectividade sem fio- Dispositivos	Não disponível	Disponível	Não disponível em todasRegiões da AWS
Conectividade sem fio- Perfis	Não disponível	Disponível	Não disponível em todasRegiões da AWS
Conectividade sem fio- Destinos	Não disponível	Disponível	Não disponível em todasRegiões da AWS
Secure- Certificados	Disponível	Disponível	
Secure- Políticas	Disponível	Disponível	
Secure- CAs	Disponível	Disponível	
Secure- Aliases de funções	Disponível	Disponível	
Secure- Autorizadores	Disponível	Disponível	
Defender- Introdução	Disponível	Sem disponibilidade no momento	
Defender- Auditoria	Disponível	Sem disponibilidade no momento	
Defender- Detecção	Disponível	Sem disponibilidade no momento	
Defender- Ações de mitigação	Disponível	Sem disponibilidade no momento	
Defender- Configurações da	Disponível	Sem disponibilidade no momento	Não disponível em todasRegiões da AWS
Aja- Regras	Disponível	Disponível	
Aja- Destinos	Disponível	Disponível	
Teste- Device Advisor	Disponível	Disponível	Não disponível em todosRegiões da AWS
Teste- Cliente de teste MQTT	Disponível	Disponível	
Software	Disponível	Sem disponibilidade no momento	
Configurações	Não disponível	Disponível	
Saiba mais	Disponível	Sem disponibilidade no momento	

Legenda

Valores de status

- Disponível

  Essa experiência de interface de usuário pode ser usada.
- Não disponível

  Essa experiência de interface de usuário não pode ser usada.
- Sem disponibilidade no momento

  A nova experiência da interface do usuário está sendo trabalhada, mas ainda não está pronta.
- Em andamento

  A nova experiência de interface de usuário está em processo de atualização. No entanto, algumas páginas ainda podem ter a experiência do usuário original.

Conceitos básicos do AWS IoT Core

Se você é novo na IoT ou tem anos de experiência, esses recursos apresentam oAWS IoTconceitos e termos que ajudarão você a começar a usarAWS IoT.

- LookdentroAWS IoTe seus componentes emComo o AWS IoT funciona (p. 3).
- Saiba maisMais sobreAWS IoT (p. 12)de nossa coleção de materiais de treinamento e vídeos. Este tópico também inclui uma lista de serviços queAWS IoTpode se conectar a links de mídia social e links para especificações de protocolo de comunicação.
- the section called “Connect o primeiro dispositivo aoAWS IoT Core” (p. 18).
- Desenvolversuas soluções de IoT porComo conectar-se ao AWS IoT Core (p. 71)e explorar oTutoriais do AWS IoT (p. 125).
- Testar e validarseus dispositivos de IoT para comunicação segura e confiável usando oDevice Advisor (p. 1067).
- Gerenciesua solução usandoAWS IoT CoreServiços de gerenciamento comoIndexação de frota (p. 836),Trabalhos (p. 681), eAWS IoT Device Defender (p. 882).
- Analisaros dados de seus dispositivos usando oServiços de dados do AWS IoT (p. 7).

Connect o primeiro dispositivo aoAWS IoT Core

AWS IoT Coreserviços conectam dispositivos IoT aAWS IoTServiços e outrosAWSServiços da .AWS IoT Coreinclui o gateway do dispositivo e o agente de mensagens, que conectam e processam mensagens entre seus dispositivos IoT e a nuvem.

Veja como você pode começar a usarAWS IoT CoreeAWS IoT.

Esta seção apresenta um tour peloAWS IoT CorePara apresentar seus principais serviços e fornecer vários exemplos de como conectar um dispositivo aoAWS IoT Coree passe mensagens entre eles. Passar mensagens entre dispositivos e a nuvem é fundamental para todas as soluções de IoT e é como seus dispositivos podem interagir com outrosAWSServiços da .

- Configurar seuConta da AWS (p. 19)

  Antes que você possa usarAWS IoTServiços, você deve configurar umConta da AWS. Se você já tem umConta da AWSe um usuário do IAM para você, você pode usá-los e pular esta etapa.

- Experimente o tutorial interativo (p. 21)

  Esta demonstração é melhor se você quiser ver o que é básicoAWS IoTpode fazer sem conectar um dispositivo ou baixar qualquer software. O tutorial interativo apresenta uma solução simulada baseadaAWS IoT Coreserviços que ilustram como eles interagem.
- Experimente o tutorial de conexão rápida (p. 24)

  Este tutorial é melhor se você quiser começar a usar rapidamenteAWS IoTe veja como funciona em um cenário limitado. Neste tutorial, você precisará de um dispositivo e instalará algunsAWS IoTsoftware nele. Se não tiver um dispositivo IoT, você poderá usar seu computador pessoal Windows, Linux ou macOS como um dispositivo neste tutorial. Se você quiser tentarAWS IoT, mas você não tem um dispositivo, tente a próxima opção.
- Explorar oAWS IoT Coreserviços com um tutorial prático (p. 38)

  Este tutorial é melhor para desenvolvedores que desejam começar a usarAWS IoTpara que eles possam continuar a explorar outrosAWS IoT Corerecursos como o mecanismo de regras e sombras. Este tutorial segue um processo semelhante ao tutorial de conexão rápida, mas fornece mais detalhes sobre cada etapa para permitir uma transição mais suave para os tutoriais mais avançados.
- Visualizar mensagens MQTT com oAWS IoTCliente MQTT (p. 66)

  Saiba como usar o cliente de teste MQTT para ver seu primeiro dispositivo publicar mensagens MQTT noAWS IoT. O cliente de teste MQTT é uma ferramenta útil para monitorar e solucionar problemas de conexões de dispositivos.

  > Note
  >
  > Se você quiser experimentar mais de um desses tutoriais de introdução ou repetir o mesmo tutorial, você deve excluir o objeto que você criou de um tutorial anterior antes de iniciar outro. Se você não excluir o objeto coisa de um tutorial anterior, precisará usar um nome diferente para tutoriais subsequentes. Isso ocorre porque o nome da coisa deve ser exclusivo na sua conta eRegião da AWS.

Para obter mais informações sobre o AWS IoT Core, consulte O que é o AWS IoT Core (p. 1)?

Configurar seuConta da AWS

Antes de usar o AWS IoT Core pela primeira vez, execute as seguintes tarefas:

- Cadastrar-se em uma Conta da AWS (p. 19)
- Crie um usuário e conceda permissões (p. 20)
- Abrir oAWS IoTconsole (p. 21)

Se você já tem umConta da AWSe um usuário do IAM para você, você pode usá-los e pular em frente parathe section called “Abrir oAWS IoTconsole” (p. 21).

Cadastrar-se em uma Conta da AWS

Quando você se inscreve paraAWS, sua conta da será automaticamente cadastrada em todos os serviços emAWS. Se você tiver umConta da AWSjá, pule este procedimento. Se você não tem uma Conta da AWS, siga o procedimento abaixo para criar uma.

Você pode esperar gastar cerca de 5 minutos configurando seuConta da AWS.

Se você ainda não tem Conta da AWS, siga as etapas a seguir para criar um.

Para se cadastrar em uma Conta da AWS

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

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

   Note

   Salve suas despesasConta da AWSnúmero, porque você precisará dele na próxima tarefa.

Crie um usuário e conceda permissões

Este procedimento descreve como criar um usuário do IAM para você e adicionar esse usuário a um grupo com permissões administrativas de uma política gerenciada anexada. O IAM é oAWSserviço que gerencia usuários e acesso aAWSrecursos da AWS. Você deve fazer isso para que você possa criar oAWS IoTrecursos em sua conta e conceda-lhes permissão para fazer o que precisam fazer.

Para criar um usuário administrador para você mesmo e adicionar o usuário a um grupo de administradores (console)

1. Faça login no console do IAM como proprietário da conta escolhendo Root user (Usuário root) e inserindo o endereço de e-mail da sua Conta da AWS. Na próxima página, insira sua senha.

   Note

   Recomendamos seguir as práticas recomendadas para utilizar o usuário do IAM **`Administrator`** a seguir e armazenar as credenciais do usuário raiz com segurança. Cadastre-se como o usuário raiz apenas para executar algumas tarefas de gerenciamento de serviços e contas.
2. No painel de navegação, escolha Users (Usuários) e Add users (Adicionar usuários).
3. Em User name (Nome do usuário), digite **`Administrator`**.
4. Marque a caixa de seleção ao lado do acesso ao AWS Management Console. Então, selecione Custom password (Senha personalizada), e insira sua nova senha na caixa de texto.
5. (Opcional) Por padrão, a AWS exige que o novo usuário crie uma senha ao fazer login pela primeira vez. Você pode desmarcar a caixa de seleção próxima de User must create a new password at next sign-in (O usuário deve criar uma senha no próximo login) para permitir que o novo usuário redefina a senha depois de fazer login.
6. Selecione Next (Próximo): Permissions
7. Em Set permissions (Conceder permissões), escolha Add user to group (Adicionar usuário ao grupo).
8. Escolha Create group (Criar grupo).
9. Na caixa de diálogo Create group (Criar grupo), em Group name (Nome do grupo), digite **`Administrators`**.
10. Escolha Filter policies (Filtrar políticas) e, depois, selecione AWS managed — job function (Função de trabalho gerenciada da AWS) para filtrar o conteúdo da tabela.
11. Na lista de políticas, marque a caixa de seleção AdministratorAccess. A seguir escolha Criar grupo.

    Note

    Você deve ativar o acesso de usuário do IAM e da função para Billing (Faturamento) antes de usar as permissões de `AdministratorAccess` para acessar o console do AWS Billing and Cost Management. Para fazer isso, siga as instruções na etapa 1 do tutorial sobre como delegar acesso ao console de faturamento.

12. Suporte a lista de grupos, selecione a caixa de seleção para seu novo grupo. Escolha Refresh (Atualizar) caso necessário, para ver o grupo na lista.
13. Selecione Next (Próximo): Tags.
14. (Opcional) Adicione metadados ao usuário anexando tags como pares de chave-valor. Para obter mais informações sobre como usar etiquetas no IAM, consulte Marcar entidades do IAM no Guia do usuário do IAM.
15. Selecione Next (Próximo): Review (Revisar)Para ver uma lista de associações de grupos a serem adicionadas ao novo usuário. Quando você estiver pronto para continuar, escolha Create user (Criar usuário).

Você pode usar esse mesmo processo para criar mais grupos e usuários e conceder aos seus usuários acesso aos recursos da sua Conta da AWS. Para saber como usar políticas para restringir as permissões de usuário a recursos específicos da AWS, consulte Gerenciamento de acesso e Exemplos de políticas.

Abrir oAWS IoTconsole

A maioria dos tópicos orientados para consoles nesta seção começa a partir doAWS IoTconsole. Se você ainda não estiver conectado aoConta da AWS, faça login e, em seguida, abra oAWS IoTconsoleE continue até a próxima seção para continuar começando comAWS IoT.

Experimente oAWS IoT Coretutorial interativo

O tutorial interativo mostra os componentes de uma solução de IoT simples baseadaAWS IoT. As animações do tutorial mostram como os dispositivos IoT interagem comAWS IoT CoreServiços da . Este tópico fornece uma visualização doAWS IoT Coretutorial interativo.

Para executar a demonstração, primeiro você devethe section called “Configurar seuConta da AWS” (p. 19). O tutorial, no entanto, não requer nenhumAWS IoTrecursos, software adicional ou qualquer codificação.

Espere gastar cerca de 5 a 10 minutos nesta demonstração. Dar a si mesmo 10 minutos permitirá mais tempo para considerar cada uma das etapas.

Para executar aAWS IoT Coretutorial interativo

1. Abrir oHub de aprendizadonoAWS IoTconsole do .

   NoBem-vindo aoAWS IoTconsoleA página, noVeja comoAWS IoTfuncionaBloco, escolhaInicie o tutorial.

2. NoAWS IoTpara “Happ, revise as partes do tutorial e escolhaIniciar o tutorialQuando estiver pronto para continuar.

As seções a seguir descrevem como oAWS IoTpara “HappApresenta estesAWS IoT CoreRecursos do :

- Conectar dispositivos IoT (p. 22)
- Salvando o estado do dispositivo offline (p. 23)
- Encaminhamento de dados do dispositivo para serviços (p. 23)

Conectar dispositivos IoT

Saiba como os dispositivos IoT se comunicam comAWS IoT Core.

A animação nesta etapa mostra como dois dispositivos, o dispositivo de controle à esquerda e uma lâmpada inteligente na casa à direita, conectam e se comunicam comAWS IoT CoreNa nuvem. A animação mostra os dispositivos se comunicando comAWS IoT Coree reagindo às mensagens que recebem.

A imagem no console inclui animações que não aparecem nesta imagem.

Para obter mais informações sobre a conexão de dispositivos aoAWS IoT Core, consulteComo conectar-se ao AWS IoT Core (p. 71).

Salvando o estado do dispositivo offline

Saiba comoAWS IoT Coresalva o estado do dispositivo enquanto um dispositivo ou aplicativo estiver offline.

A animação nesta etapa mostra como o serviço Device Shadow noAWS IoT Coresalva informações de estado do dispositivo para o dispositivo de controle e a lâmpada inteligente. Enquanto a lâmpada inteligente está offline, o Device Shadow salva comandos do dispositivo de controle.

Quando a lâmpada inteligente se reconecta aoAWS IoT Core, ele recupera esses comandos. Quando o dispositivo de controle está offline, o Device Shadow salva informações de estado da lâmpada inteligente. Quando o dispositivo de controle se reconecta, ele recupera o estado atual da lâmpada inteligente para atualizar sua exibição.

Para obter mais informações sobre as Device Shadows, consulteServiço Device Shadow do AWS IoT (p. 633).

Encaminhamento de dados do dispositivo para serviços

Saiba comoAWS IoT Coreenvia o estado do dispositivo para outrosAWSServiços da .

A animação nesta etapa mostra comoAWS IoT Coreenvia dados dos dispositivos para outrosAWSServiços usando oAWS IoTregras.AWS IoTas regras assinam mensagens específicas dos dispositivos, interpretam os dados nessas mensagens e roteiam os dados interpretados para outros serviços. Neste exemplo, umAWS IoTrule interpreta dados de um sensor de movimento e envia comandos para um Device Shadow, que os envia para a lâmpada inteligente. Como no exemplo anterior, o Device Shadow armazena as informações do estado do dispositivo para o dispositivo de controle.

Para obter mais informações sobre regras de AWS IoT, consulte Regras para AWS IoT (p. 478).

Experimente oAWS IoTConexão rápida

Neste tutorial, você criará seu objeto de primeira coisa, conectará um dispositivo a ele e o verá enviar mensagens MQTT.

Você pode esperar gastar 15-20 minutos neste tutorial.

Este tutorial é melhor para pessoas que desejam começar rapidamente comAWS IoTpara ver como ele funciona em um cenário limitado. Se você está procurando um exemplo que o fará começar para que você possa explorar mais recursos e serviços, tenteExplorar oAWS IoT Coreserviços em hands-on tutorial (p. 38).

Neste tutorial, você baixará e executará software em um dispositivo que se conecta a umRecurso de coisasemAWS IoT Corecomo parte de uma solução IoT muito pequena. O dispositivo pode ser um dispositivo IoT, como um Raspberry Pi, ou também pode ser um computador que esteja executando Linux, OS e OSX ou Windows. Se você deseja conectar um dispositivo WAN de longo alcance (LoRaWAN) aAWS IoT, consulte o tutorialConectando dispositivos e gateways aAWS IoT Corepara LoRaWAN (p. 1126).

Se o seu dispositivo suportar um navegador que possa executar oAWS IoTconsole, recomendamos que você conclua este tutorial nesse dispositivo.

> **Note**
>
> Se o dispositivo não tiver um navegador compatível, siga este tutorial em um computador. Quando o procedimento solicitar que você baixe o arquivo, faça o download para o computador e transfira o arquivo baixado para o seu dispositivo usando o Secure Copy (SCP) ou um processo semelhante.

O tutorial requer que seu dispositivo IoT se comunique com a porta 8443 em seuConta da AWSendpoint de dados do dispositivo. Para testar se ele pode acessar essa porta, experimente os procedimentos emTestando a conectividade com o endpoint de dados do dispositivo (p. 34).

Etapa 1. Inicie o tutorial

Se possível, conclua este procedimento em seu dispositivo; caso contrário, esteja pronto para transferir um arquivo para o dispositivo posteriormente neste procedimento.

1. Open YourAWS IoTconsoleE, no menu esquerdo, escolhaSaiba mais.

2. NoConectar-se aoAWS IoTBloco, escolhaVisualizar as opções de conexão.

3. NoInvocar um dispositivoBloco, escolhaConceitos básicos.

4. Revise as etapas que descrevem o que você fará neste tutorial. Quando estiver pronto para continuar, escolhaConceitos básicos.

Etapa 2. Criar um objeto de coisa

1. NoComo você está se conectando comAWSIoT?, escolha a plataforma e o idioma doAWS IoTSDK do dispositivo que você deseja usar. Este exemplo usa a plataforma Linux/OSX e o Python SDK. Certifique-se de ter python3 e pip3 instalados no dispositivo de destino antes de prosseguir para a próxima etapa.

   Note

   Certifique-se de verificar a lista de softwares de pré-requisitos exigidos pelo SDK escolhido na parte inferior da página do console.
   É necessário ter o software necessário instalado no computador de destino antes de prosseguir para a próxima etapa.

   Depois de escolher o idioma SDK da plataforma e do dispositivo, escolhaPróximo.

2. NoName (Nome), insira o nome para o objeto da coisa. O nome usado neste exemplo é**`MyIotThing`**.

   Important

   Verifique novamente o nome da sua coisa antes de continuar.
   O nome de uma coisa não pode ser alterado depois que o objeto da coisa é criado. Se você quiser alterar um nome de coisa, você deve criar um objeto de coisa nova com o nome correto da coisa e, em seguida, excluir aquele com o nome incorreto.

3. Depois de dar um nome ao objeto coisa, escolhaPróxima etapa.

Etapa 3. Faça download de arquivos para o dispositivo

Esta página é exibida depoisAWS IoTcriou o kit de conexão, que inclui os seguintes arquivos e recursos que seu dispositivo requer:

- Os arquivos de certificado da coisa usados para autenticar o dispositivo
- Um recurso de política para autorizar seu objeto coisa a interagir comAWS IoT
- O script para baixar oAWSSDK de dispositivo e execute o programa de exemplo no seu dispositivo

1. Quando estiver pronto para continuar, escolha oBaixe o kit de conexão parapara baixar o kit de conexão para a plataforma que você escolheu anteriormente.

2. Se você estiver executando esse procedimento em seu dispositivo, salve o arquivo do kit de conexão em um diretório no qual você pode executar comandos de linha de comando.

   Se você não estiver executando esse procedimento no dispositivo, salve o arquivo do kit de conexão em um diretório local e transfira o arquivo para o dispositivo.

3. Depois de ter o arquivo do kit de conexão no dispositivo, continue o tutorial escolhendoPróxima etapa.

Etapa 4. Executar a amostra

Você faz esse procedimento em um terminal ou janela de comando em seu dispositivo enquanto segue as instruções exibidas no console. Os comandos que você vê no console são para o sistema operacional que você escolheu nothe section called “Etapa 2. Criar um objeto de coisa” (p. 27). Aqueles mostrados aqui são para os sistemas operacionais Linux/OSX.

1. Em um terminal ou janela de comando em seu dispositivo, no diretório com o arquivo do kit de conexão, execute as etapas mostradas naAWS IoTconsole do .

   Se você estiver usando um Windows PowerShell janela de comando e ounzipcomando não funciona, substituaunzipcomexpand-archivee tente a linha de comando novamente.

2. Depois de digitar o comando deEtapa 3no console, você verá uma saída no terminal ou na janela de comando do dispositivo semelhante à seguinte. Esta saída é das mensagens para as quais o programa está enviando e depois recebendo de voltaAWS IoT Core.

```
Received a new message:
{"message": "Hello World!", "sequence": 1}
from topic:
sdk/test/Python
--------------


2021-12-18 00:19:00,752 - AWSIoTPythonSDK.core.protocol.internal.clients - DEBUG - Invoking custom event callback...
2021-12-18 00:19:01,757 - AWSIoTPythonSDK.core.protocol.mqtt_core - INFO - Performing sync publish...
2021-12-18 00:19:01,757 - AWSIoTPythonSDK.core.protocol.internal.clients - DEBUG - Filling in custom puback (QoS>0) event callback...
2021-12-18 00:19:01,846 - AWSIoTPythonSDK.core.protocol.internal.workers - DEBUG - Produced [puback] event
2021-12-18 00:19:01,847 - AWSIoTPythonSDK.core.protocol.internal.workers - DEBUG - Dispatching [puback] event
2021-12-18 00:19:01,847 - AWSIoTPythonSDK.core.protocol.internal.clients - DEBUG - Invoking custom event callback...
2021-12-18 00:19:01,847 - AWSIoTPythonSDK.core.protocol.internal.clients - DEBUG - This custom event callback is for pub/sub/unsub, removing it after invocation...
2021-12-18 00:19:01,871 - AWSIoTPythonSDK.core.protocol.internal.workers - DEBUG - Produced [message] event
2021-12-18 00:19:01,875 - AWSIoTPythonSDK.core.protocol.internal.workers - DEBUG - Dispatching [message] event
Received a new message:
{"message": "Hello World!", "sequence": 2}
from topic:
sdk/test/Python
--------------


2021-12-18 00:19:01,875 - AWSIoTPythonSDK.core.protocol.internal.clients - DEBUG - Invoking custom event callback...
2021-12-18 00:19:02,871 - AWSIoTPythonSDK.core.protocol.mqtt_core - INFO - Performing sync publish...
2021-12-18 00:19:02,871 - AWSIoTPythonSDK.core.protocol.internal.clients - DEBUG - Filling in custom puback (QoS>0) event callback...
2021-12-18 00:19:02,956 - AWSIoTPythonSDK.core.protocol.internal.workers - DEBUG - Produced [puback] event
2021-12-18 00:19:02,960 - AWSIoTPythonSDK.core.protocol.internal.workers - DEBUG - Dispatching [puback] event
2021-12-18 00:19:02,960 - AWSIoTPythonSDK.core.protocol.internal.clients - DEBUG - Invoking custom event callback...
2021-12-18 00:19:02,960 - AWSIoTPythonSDK.core.protocol.internal.clients - DEBUG - This custom event callback is for pub/sub/unsub, removing it after invocation...
2021-12-18 00:19:02,983 - AWSIoTPythonSDK.core.protocol.internal.workers - DEBUG - Produced [message] event
2021-12-18 00:19:02,987 - AWSIoTPythonSDK.core.protocol.internal.workers - DEBUG - Dispatching [message] event
Received a new message:
{"message": "Hello World!", "sequence": 3}
from topic:
sdk/test/Python
--------------
```

Enquanto o programa de amostra está em execução, emEtapa 4: Enviar uma mensagem para o dispositivo, insira uma mensagem como`Hello World!`noAWS IoTconsole do . Para enviar a mensagem, escolhaEnvie-me. A mensagem de teste aparece no terminal ou na janela de comando do dispositivo.

> **Note**
>
> Para obter mais informações sobre assinatura e publicação de tópicos, consulte o código de exemplo do SDK escolhido.

3. Para executar o programa de exemplo novamente, você pode repetir os comandos doEtapa 3/3no console deste procedimento.
4. (Opcional) Se quiser ver as mensagens do cliente de IoT noAWS IoTconsole, abra oCliente de teste MQTTnoTesteA página doAWS IoTconsole do . Se você escolheu o Python SDK, então noCliente de teste MQTT, emFiltro de tópicos, insira o tópico, como**`sdk/test/Python`**para assinar as mensagens do seu dispositivo. Os filtros de tópicos diferenciam maiúsculas e minúsculas e dependem da linguagem de programação do SDK que você escolheuEtapa 1/1. Para obter mais informações sobre assinatura e publicação de tópicos, consulte o exemplo de código do SDK escolhido.
5. Depois de assinar o tópico de teste, execute./start.shno dispositivo. Para obter mais informações, consulte the section called “Visualizar mensagens MQTT com o cliente MQTT do AWS IoT” (p. 66).

   Depois de correr./start.sh, mensagens aparecem no cliente MQTT, semelhante ao seguinte:

```
{
  "message": "Hello World!",
  "sequence": 10
}
```

   O`sequence`incrementos numéricos em um cada vez que um novo`Hello World`mensagem é recebida e pára quando você encerra o programa.
6. Para finalizar o tutorial e ver um resumo, noAWS IoTConsole do, escolhaConcluído.

Etapa 5. Explorar ainda mais

Aqui estão algumas ideias para explorarAWS IoTmais depois de concluir o início rápido.

- Visualizar mensagens MQTT no cliente MQTT

DoAWS IoTconsole, você pode abrir oCliente MQTTnoTesteA página doAWS IoTconsole do . NoCliente MQTT, assine no#E, em seguida, no dispositivo, execute o programa./start.shComo descrito na etapa anterior. Para obter mais informações, consulte the section called “Visualizar mensagens MQTT com o cliente MQTT do AWS IoT” (p. 66).

- Execute testes em seus dispositivos comDevice Advisor

  Use o Device Advisor para testar se seus dispositivos podem se conectar de forma segura e confiável e interagir comAWS IoT.

- the section called “Experimente oAWS IoT Coretutorial interativo” (p. 21)

  Para iniciar o tutorial interativo, a partir doSaiba maisA página doAWS IoTConsole do, noVeja comoAWS IoTfuncionaBloco, escolhaIniciar o tutorial.

- Prepare-se para explorar mais tutoriais (p. 38)

  Este início rápido oferece apenas uma amostra deAWS IoT. Se você quiser explorarAWS IoTmais e saiba mais sobre os recursos que a tornam uma poderosa plataforma de soluções de IoT, comece a preparar sua plataforma de desenvolvimentoExplorar oAWS IoT Coreserviços em hands-on tutorial (p. 38).

Testando a conectividade com o endpoint de dados do dispositivo

Este tópico descreve como testar a conexão de um dispositivo com aPonto de extremidade de dados de dispositivo, o endpoint que seus dispositivos IoT usam para se conectarAWS IoT.

Execute esses procedimentos no dispositivo que você deseja testar ou usando uma sessão de terminal SSH conectada ao dispositivo que você deseja testar.

Para testar a conectividade de um dispositivo com o endpoint de dados do dispositivo.

- Encontre o endpoint de dados do dispositivo (p. 34)
- Testar a conexão rapidamente (p. 34)
- Faça com que o aplicativo teste a conexão com o endpoint e a porta de dados do dispositivo (p. 35)
- Teste a conexão com o endpoint e a porta de dados do dispositivo (p. 37)

Encontre o endpoint de dados do dispositivo

Para localizar o endpoint de dados do dispositivo

1. NoAWS IoTconsole, perto da parte inferior do painel de navegação, escolhaConfigurações.
2. NoConfigurações A página, noPonto de extremidade de dados de dispositivocontêiner, localize oPonto finalvalor e copie-o.
3. Seu valor de endpoint é exclusivo para oConta da AWSE é semelhante a este exemplo:`a3qEXAMPLEsffp-ats.iot.eu-west-1.amazonaws.com`.

Salve o endpoint de dados do dispositivo para usar nos procedimentos a seguir.

Testar a conexão rapidamente

Este procedimento testa a conectividade geral com o endpoint de dados do dispositivo, mas não testa a porta específica que seus dispositivos usarão. Este teste usa um programa comum e geralmente é suficiente para saber se seus dispositivos podem se conectar aoAWS IoT.

Se você quiser testar a conectividade com a porta específica que seus dispositivos usarão, pule este procedimento e continue aFaça com que o aplicativo teste a conexão com o endpoint e a porta de dados do dispositivo (p. 35).

Para testar o endpoint de dados do dispositivo rapidamente

1. Em uma janela de terminal ou linha de comando no dispositivo, substitua o ponto final de dados do dispositivo de amostra (*a3qEXAMPLEsffp-ats.iot.eu-west-1.amazonaws.com*) com o endpoint de dados do dispositivo para sua conta e, em seguida, insira este comando.

   Linux

   ```
   ping -c 5 a3qEXAMPLEsffp-ats.iot.eu-west-1.amazonaws.com
   ```

   Windows

   ```
   ping -n 5 a3qEXAMPLEsffp-ats.iot.eu-west-1.amazonaws.com
   ```

2. Se`ping`exibe uma saída semelhante à seguinte, conectada ao endpoint de dados do dispositivo com êxito. Embora não tenha se comunicado comAWS IoTdiretamente, ele encontrou o servidor e é provável queAWS IoTestá disponível por meio desse endpoint.

   ```
   PING a3qEXAMPLEsffp-ats.iot.eu-west-1.amazonaws.com (xx.xx.xxx.xxx) 56(84) bytes of
    data.
   64 bytes from ec2-EXAMPLE-218.eu-west-1.compute.amazonaws.com (xx.xx.xxx.xxx):
    icmp_seq=1 ttl=231 time=127 ms
   64 bytes from ec2-EXAMPLE-218.eu-west-1.compute.amazonaws.com (xx.xx.xxx.xxx):
    icmp_seq=2 ttl=231 time=127 ms
   64 bytes from ec2-EXAMPLE-218.eu-west-1.compute.amazonaws.com (xx.xx.xxx.xxx):
    icmp_seq=3 ttl=231 time=127 ms
   64 bytes from ec2-EXAMPLE-218.eu-west-1.compute.amazonaws.com (xx.xx.xxx.xxx):
    icmp_seq=4 ttl=231 time=127 ms
   64 bytes from ec2-EXAMPLE-218.eu-west-1.compute.amazonaws.com (xx.xx.xxx.xxx):
    icmp_seq=5 ttl=231 time=127 ms
   ```

   Se estiver satisfeito com esse resultado, poderá parar de testar aqui.

   Se você quiser testar a conectividade com a porta específica usada porAWS IoT, continue aFaça com que o aplicativo teste a conexão com o endpoint e a porta de dados do dispositivo (p. 35).

3. Se`ping`não retornou uma saída bem-sucedida, verifique o valor do endpoint para garantir que você tenha o endpoint correto e verifique a conexão do dispositivo com a Internet.

Faça com que o aplicativo teste a conexão com o endpoint e a porta de dados do dispositivo

Um teste de conectividade mais completo pode ser realizado usando`nmap`. Este procedimento testa para ver se`nmap`Está instalado no dispositivo.

Para verificar se**`nmap`**No dispositivo

1. Em uma janela de terminal ou linha de comando no dispositivo que você deseja testar, digite este comando para ver se`nmap`está instalado.

   ```
   nmap --version
   ```

2. Se você vir uma saída semelhante à seguinte,`nmap`Está instalado e você pode prosseguir parathe section called “Teste a conexão com o endpoint e a porta de dados do dispositivo” (p. 37).

```
Nmap version 6.40 ( http://nmap.org )
Platform: x86_64-koji-linux-gnu
Compiled with: nmap-liblua-5.2.2 openssl-1.0.2k libpcre-8.32 libpcap-1.5.3 nmap-
libdnet-1.12 ipv6
Compiled without:
Available nsock engines: epoll poll select
```

3. Se não vir uma resposta semelhante à mostrada na etapa anterior, instale`nmap`No dispositivo. Selecione o procedimento para o sistema operacional do seu dispositivo.

Linux

Esse procedimento requer que você tenha permissão para instalar o software no computador.

Para instalar o nmap em seu computador Linux

1. Em uma janela de terminal ou linha de comando em seu dispositivo, insira o comando que corresponde à versão do Linux que está executando.

   a. Debian ou Ubuntu:

   ```
   sudo apt install nmap
   ```

   b. CentOS ou RHEL:

   ```
   sudo yum install nmap
   ```

2. Teste a instalação com este comando:

   ```
   nmap --version
   ```

3. Se você vir uma saída semelhante à seguinte,`nmap`Está instalado e você pode prosseguir parathe section called “Teste a conexão com o endpoint e a porta de dados do dispositivo” (p. 37).

   ```
   Nmap version 6.40 ( http://nmap.org )
   Platform: x86_64-koji-linux-gnu
   Compiled with: nmap-liblua-5.2.2 openssl-1.0.2k libpcre-8.32 libpcap-1.5.3 nmap-
   libdnet-1.12 ipv6
   Compiled without:
   Available nsock engines: epoll poll select
   ```

macOS

Esse procedimento requer que você tenha permissão para instalar o software no computador.

Para instalar o nmap em seu computador macOS

1. Em um navegador, abrahttps://nmap.org/download#macosxe faça download doestável mais recenteinstalador.

   Quando solicitado, selecioneAberto com o DiskImageInstaller.
2. Na janela de instalação, mova o pacote para oAplicativosfolder.
3. NoLocalizador, localize `onmap-xxxx-mpkg`pacote noAplicativosfolder.Ctrl-clicko pacote on e selecioneAbertopara abrir o pacote.

4. Revise a caixa de diálogo de segurança. Se você estiver pronto para instalarnmap, escolhaAbertopara instalarnmap.
5. DentroTerminal, teste a instalação com este comando.

```
nmap --version
```

6. Se você vir uma saída semelhante à seguinte,`nmap`Está instalado e você pode prosseguir parathe section called “Teste a conexão com o endpoint e a porta de dados do dispositivo” (p. 37).

```
Nmap version 7.92 ( https://nmap.org )
Platform: x86_64-apple-darwin17.7.0
Compiled with: nmap-liblua-5.3.5 openssl-1.1.1k nmap-libssh2-1.9.0 libz-1.2.11
 nmap-libpcre-7.6 nmap-libpcap-1.9.1 nmap-libdnet-1.12 ipv6 Compiled without:
Available nsock engines: kqueue poll select
```

Windows

Esse procedimento requer que você tenha permissão para instalar o software no computador.

Para instalar o nmap no computador Windows

1. Em um navegador, abrahttps://nmap.org/download#windowse faça download doestável mais recenteLançamento do programa de configuração.

   Caso solicitado, escolhaSalvar arquivo. Depois que o arquivo for baixado, abra-o na pasta de downloads.
2. Depois que o arquivo de configuração terminar o download, abra baixadonmap-xxxx-setup.exePara instalar o aplicativo.
3. Aceite as configurações padrão à medida que o programa for instalado.

   Você não precisa do aplicativo Npcap para este teste. Você pode cancelar a seleção dessa opção se não desejar instalá-la.
4. DentroCommand, teste a instalação com este comando.

```
nmap --version
```

5. Se você vir uma saída semelhante à seguinte,`nmap`Está instalado e você pode prosseguir parathe section called “Teste a conexão com o endpoint e a porta de dados do dispositivo” (p. 37).

```
Nmap version 7.92 ( https://nmap.org )
Platform: i686-pc-windows-windows
Compiled with: nmap-liblua-5.3.5 openssl-1.1.1k nmap-libssh2-1.9.0 nmap-libz-1.2.11
 nmap-libpcre-7.6 Npcap-1.50 nmap-libdnet-1.12 ipv6
Compiled without:
Available nsock engines: iocp poll select
```

Teste a conexão com o endpoint e a porta de dados do dispositivo

Para testar o endpoint e a porta de dados do dispositivo

1. Em uma janela de terminal ou linha de comando no dispositivo, substitua o ponto final de dados do dispositivo de amostra (*`a3qEXAMPLEsffp-ats.iot.eu-west-1.amazonaws.com`*) com o endpoint de dados do dispositivo para sua conta e, em seguida, insira este comando.

```
nmap -p 8443 a3qEXAMPLEsffp-ats.iot.eu-west-1.amazonaws.com
```

2. Se`nmap`exibe uma saída semelhante à seguinte:`nmap`foi capaz de se conectar com êxito ao endpoint de dados do dispositivo na porta selecionada.

```
Starting Nmap 7.92 ( https://nmap.org ) at 2022-02-18 16:23 Pacific Standard Time
Nmap scan report for a3qEXAMPLEsffp-ats.iot.eu-west-1.amazonaws.com (xx.xxx.147.160)
Host is up (0.036s latency).
Other addresses for a3qEXAMPLEsffp-ats.iot.eu-west-1.amazonaws.com (not scanned):
 xx.xxx.134.144 xx.xxx.55.139 xx.xxx.110.235 xx.xxx.174.233 xx.xxx.74.65 xx.xxx.122.179
 xx.xxx.127.126
rDNS record for xx.xxx.147.160: ec2-EXAMPLE-160.eu-west-1.compute.amazonaws.com

PORT     STATE SERVICE
8443/tcp open  https-alt
MAC Address: 00:11:22:33:44:55 (Cimsys)

Nmap done: 1 IP address (1 host up) scanned in 0.91 seconds
```

3. Se`nmap`não retornou uma saída bem-sucedida, verifique o valor do endpoint para garantir que você tenha o endpoint correto e verifique a conexão do dispositivo com a Internet.

Você pode testar outras portas no endpoint de dados do dispositivo, como a porta 443, a porta HTTPS primária, substituindo a porta usada na etapa 1,**`8443`**, com a porta que você deseja testar.

Explorar oAWS IoT Coreserviços em hands-on tutorial

Neste tutorial, você instalará o software e criará oAWS IoTrecursos necessários para conectar um dispositivo aAWS IoTpara que ele possa enviar e receber mensagens MQTT comAWS IoT Core. Você verá as mensagens no cliente MQTT naAWS IoTconsole do .

Você pode esperar gastar 20 a 30 minutos neste tutorial. Se você estiver usando um dispositivo IoT ou um Raspberry Pi, este tutorial pode demorar mais se, por exemplo, você precisar instalar o sistema operacional e configurar o dispositivo.

Este tutorial é melhor para desenvolvedores que desejam começar a usarAWS IoTpara que eles possam continuar a explorar recursos mais avançados, como o mecanismo de regras e as sombras.

Este tutorial prepara você para continuar aprendendo sobreAWS IoT Coree como ele interage com outrosAWSserviços explicando as etapas com mais detalhes do que o tutorial de início rápido. Se você estiver procurando apenas um rápido,Hello World, experimente, experimente oExperimente oAWS IoTConexão rápida (p. 24).

Depois de configurar oConta da AWSeAWS IoTconsole, você seguirá estas etapas para ver como conectar um dispositivo e fazer com que ele envie mensagens paraAWS IoT.

Next steps (Próximas etapas)

- Escolha qual opção de dispositivo é a melhor para você (p. 39)
- the section called “CriarAWS IoTrecursos” (p. 39)se você não for criar um dispositivo virtual com o Amazon EC2.
- the section called “Configurar o dispositivo” (p. 43)
- the section called “Visualizar mensagens MQTT com o cliente MQTT do AWS IoT” (p. 66)

Para obter mais informações sobre o AWS IoT Core, consulte O que é o AWS IoT Core (p. 1)?

Qual opção de dispositivo é a melhor para você?

Se você não tiver certeza de qual opção escolher, use a seguinte lista das vantagens e desvantagens de cada opção para ajudá-lo a decidir qual é a melhor para você.

Opção	Essa pode ser uma boa opção se:	Isso pode não ser uma boa opção se:
the section called “Criar um dispositivo virtual com o Amazon EC2” (p. 43)	• Você não tem seu próprio dispositivo para testar. • Você não deseja instalar nenhum software em seu próprio sistema. • Você quer testar em um sistema operacional Linux.	• Você não se sente confortável usando comandos de linha de comando. • Você não deseja incorrer em nenhum adicionalAWSCobranças. • Você não quer testar em um sistema operacional Linux.
the section called “Use seu PC ou Mac Windows ou Linux como umAWS IoTdispositivo” (p. 51)	• Você não deseja incorrer em nenhum adicionalAWSCobranças. • Você não deseja configurar nenhum dispositivo adicional.	• Você não deseja instalar nenhum software no computador pessoal. • Você quer uma plataforma de teste mais representativa.
the section called “Connect um Raspberry Pi ou outro dispositivo” (p. 57)	• Você quer testarAWS IoTcom um dispositivo real. • Você já tem um dispositivo para testar. • Você tem experiência na integração de hardware em sistemas.	• Você não quer comprar ou configurar um dispositivo apenas para experimentá-lo. • Você quer testarAWS IoTo mais simples possível, por enquanto.

CriarAWS IoTrecursos

Neste tutorial, você criará oAWS IoTrecursos que um dispositivo precisa para se conectarAWS IoTe troque mensagens.

1. Criar umAWS IoTdocumento de política, que autorizará seu dispositivo a interagir comAWS IoTServiços da .
2. Criar um objeto de coisas noAWS IoTe seu certificado de dispositivo X.509 e, em seguida, anexe o documento de política. O objeto coisa é a representação virtual do seu dispositivo naAWS IoTRegistro. O certificado autentica seu dispositivo paraAWS IoT Coree o documento de política autoriza seu dispositivo a interagir comAWS IoT.

   Note

   Se você está planejandothe section called “Criar um dispositivo virtual com o Amazon EC2” (p. 43), você pode pular esta página e continuar athe section called “Configurar o dispositivo” (p. 43). Você criará esses recursos ao criar sua coisa virtual.

Este tutorial usa oAWS IoTconsole para criar oAWS IoTrecursos da AWS. Se o seu dispositivo suportar um navegador da Web, talvez seja mais fácil executar esse procedimento no navegador da Web do dispositivo, pois você poderá baixar os arquivos de certificado diretamente para o seu dispositivo. Se você executar esse procedimento em outro computador, precisará copiar os arquivos de certificado para o dispositivo antes que eles possam ser usados pelo aplicativo de amostra.

Criar uma política do AWS IoT

Os dispositivos usam um certificado X.509 para se autenticar comAWS IoT Core. O certificado temAWS IoTpolíticas anexadas a ele. Essas políticas determinam quaisAWS IoTOperações, como se inscrever em tópicos MQTT ou publicar em tópicos MQTT, o dispositivo tem permissão para executar. O dispositivo apresenta o certificado quando se conecta e envia mensagens paraAWS IoT Core.

Siga as etapas para criar uma política que permita que seu dispositivo execute oAWS IoTOperações necessárias para executar o programa de exemplo. Você deve criar a açãoAWS IoTantes que você possa anexá-lo ao certificado do dispositivo, que você criará mais tarde.

Para criar uma política do AWS IoT

1. No menu esquerdo, escolhaSecuree, depois, escolhaPolíticas. NoVocê não tem uma política aindapage, escolhaCriar política.

   Se sua conta tiver políticas existentes, escolhaCriar.

2. NoCriar política:

   1. NoPropriedades da políticana seçãoNome da política, insira um nome para a política do (por exemplo,**`My_Iot_Policy`**). Não use informações de identificação pessoal nos nomes de política.
   2. NoDocumento de política, crie as declarações de política que concedem ou negam acesso aos recursosAWS IoT Coreoperações. Para criar uma declaração de política que conceda a todos os clientes para executar**`iot:Connect`**, siga estas etapas:
      - NoEfeito de políticascampo, escolhaPermitir. Isso permite que todos os clientes que tenham essa política anexada ao certificado executem a ação listada naAção de políticascampo.
      - NoAção de políticas, escolha uma ação de política, como**`iot:Connect`**. Ações de política são as ações que seu dispositivo precisa de permissão para executar quando executa o programa de exemplo a partir do Device SDK.
      - NoRecurso de políticacampo, insira um recurso Amazon Resource Name (ARN) ou**`*`**. UMA**`*`**para selecionar qualquer cliente (dispositivo).

      Para criar as declarações de política para**`iot:Receive`**,**`iot:Publish`**, e**`iot:Subscribe`**, escolhaAdicionar nova instruçãoe repita as etapas.

      > **Note**
      >
      > Neste início rápido, o caractere curinga (*) é usado para simplificar. Para maior segurança, você deve restringir quais clientes (dispositivos) podem se conectar e publicar mensagens especificando um ARN de cliente em vez do caractere curinga como o recurso. Os ARNs de cliente seguem este formato:`arn:aws:iot:`*`your-region`*`:`*`your-aws-account`*`:client/`*`my-client-id`*.
      > No entanto, primeiro você deve criar o recurso (como um dispositivo cliente ou sombra) antes de atribuir o ARN a uma política. Para obter mais informações, consulteAWS IoT CoreRecursos de ação do.

3. Após inserir as informações da política, selecioneCriar.

Para obter mais informações, consulte Como o AWS IoT funciona com o IAM (p. 388).

Criar um objeto de coisa

Dispositivos conectados aoAWS IoTsão representados porobjetos da coisa danoAWS IoTRegistro. UMAthing objectrepresenta um dispositivo específico ou uma entidade lógica. Ela pode ser um dispositivo físico ou sensor (por exemplo, uma lâmpada ou um interruptor de luz na parede). Ela também pode ser uma entidade lógica, como uma instância de um aplicativo, ou uma entidade física que não se conecte àAWS IoT, mas está relacionado a outros dispositivos que se conectam (por exemplo, um carro que tenha sensores do motor ou um painel de controle).

Para criar uma coisa noAWS IoTconsole

1. NoAWS IoTconsoleNo menu esquerdo, escolhaGerenciee, depois, escolhaCoisas.
2. NoCoisaspage, escolhaCriar coisas.
3. NoCriar coisaspage, escolhaCriar uma única coisae, depois, escolhaPróximo.
4. NoEspecificar as propriedades da coisa, paraNome da coisa, insira um nome para o seu negócio, como**`MyIotThing`**.

   Ao nomear coisas, escolha o nome com cuidado, porque você não pode mudar um nome de coisa depois de criá-la.

   Para alterar o nome de uma coisa, é necessário criar uma coisa, fornecer o novo nome e, depois, excluir a coisa antiga.

   > **Note**
   >
   > Não use informações de identificação pessoal no nome da sua coisa. O nome da coisa pode aparecer em comunicações e relatórios não criptografados.
5. Mantenha o restante dos campos nesta página vazios. Escolha Next (Próximo).
6. NoConfigurar certificado de dispositivo -opcionaispage, escolhaGerar automaticamente um novo certificado (recomendado). Escolha Next (Próximo).
7. NoAnexar políticas ao certificado -opcionais, selecione a política criada na seção anterior. Nessa seção, a política foi nomeada,**`My_Iot_Policy`**. SelecioneCriar coisa da.
8. NoBaixar certificados e chaves:

   1. Baixe cada um dos arquivos de certificado e chave e salve-os para mais tarde. Você precisará instalar esses arquivos em seu dispositivo.

      Quando você salva seus arquivos de certificado, forneça os nomes na tabela a seguir. Esses são os nomes de arquivos usados em exemplos posteriores.

      #### Nomes de arquivos de certificado

Arquivo	Caminho do arquivo
Chave privada	`private.pem.key`
Chave pública	(não usado nesses exemplos)
Certificado de dispositivo	`device.pem.crt`
Certificado CA raiz	`Amazon-root-CA-1.pem`
   2. Faça o download do arquivo CA raiz desses arquivos escolhendo oBaixarlink do arquivo de certificado de CA raiz que corresponde ao tipo de endpoint de dados e conjunto de cifras que você está usando. Neste tutorial, escolhaBaixarà direita deChave RSA de 2048 bits: Amazon Root CA 1e faça download doChave RSA de 2048 bits: Amazon Root CA 1Arquivo de certificado.

      > **Important**
      >
      > Você deve salvar os arquivos de certificado antes de sair desta página. Depois de sair desta página no console, você não terá mais acesso aos arquivos de certificado.
      > Se você esqueceu de baixar os arquivos de certificado criados nesta etapa, você deve sair desta tela do console, ir para a lista de coisas no console, excluir o objeto que você criou e reinicie esse procedimento desde o início.
   3. Escolha Done (Concluído).

Depois de concluir este procedimento, você verá o objeto novo na sua lista de coisas.

Configurar o dispositivo

Esta seção descreve como configurar o dispositivo para se conectar aoAWS IoT. Se quiser começar a usarAWS IoTmas ainda não tem um dispositivo, você pode criar um dispositivo virtual usando o Amazon EC2 ou usar seu PC Windows ou Mac como um dispositivo IoT.

Selecione a melhor opção de dispositivo para você experimentarAWS IoT. Claro, você pode experimentar todos eles, mas tente apenas um de cada vez. Se não tiver certeza qual opção de dispositivo é melhor para você, leia sobre como escolherqual opção de dispositivo é a melhor (p. 39)E depois retorne a esta página.

Opções de dispositivo

- Criar um dispositivo virtual com o Amazon EC2 (p. 43)
- Use seu PC ou Mac Windows ou Linux como umAWS IoTdispositivo (p. 51)
- Connect um Raspberry Pi ou outro dispositivo (p. 57)

Criar um dispositivo virtual com o Amazon EC2

Neste tutorial, você criará uma instância do Amazon EC2 para servir como seu dispositivo virtual na nuvem.

Para concluir este tutorial, você precisará de umConta da AWS. Se você ainda não tiver um, conclua as etapas descritas emConfigurar seuConta da AWS (p. 19)Antes de continuar.

Neste tutorial, você irá:

- Configurar uma instância do Amazon EC2 (p. 43)
- Instale o Git, Node.js e configure oAWS CLI (p. 44)
- CriarAWS IoTrecursos para seu dispositivo virtual (p. 46)
- Instalar o SDK de dispositivos da AWS IoT para JavaScript (p. 49)
- Execute o aplicativo de amostra (p. 49)
- Exibir mensagens do aplicativo de amostra naAWS IoTconsole (p. 50)

Configurar uma instância do Amazon EC2

As etapas a seguir mostram como criar uma instância do Amazon EC2 que funcionará como seu dispositivo virtual no lugar de um dispositivo físico.

Se esta for a primeira vez que você cria uma instância do Amazon EC2, você poderá encontrar as instruções noComece a usar as instâncias do Amazon EC2Linuxpara ser mais útil.

Como iniciar uma instância

1. Abra o console do Amazon EC2 em https://console.aws.amazon.com/ec2/.
2. No painel do console, selecione Launch Instance.
3. OEtapa 1: Escolha uma Imagem de máquina da Amazon (AMI)A página exibe uma lista de configurações básicas, chamadasImagens de máquina da Amazon (AMIs), que servem como modelos para sua instância. Selecione uma versão HVM do Amazon Linux 2,comoAmazon Linux 2 AMI (HVM), SSD Volume Type. Observe que essas AMIs estão marcadas como “Free tier eligible” (Qualificáveis para nível gratuito).
4. Na página Choose an Instance Type, é possível selecionar a configuração de hardware de sua instância. Selecione o tipo `t2.micro`, que é selecionado por padrão. Observe que este tipo de instância está qualificado para o nível gratuito.

5. Escolha Review and Launch para permitir que o assistente conclua outras definições de configuração para você.
6. Na página Review Instance Launch, escolha Launch.
7. Quando solicitado um key pair, selecioneCreate a new key pair, insira um nome para o key pair e escolhaFazer download do par de chaves. Esta é sua única chance de salvar o arquivo de chave privada, então, certifique-se de fazer download dele. Salve o arquivo de chave privada em um lugar seguro. Você precisará fornecer o nome do par de chaves ao iniciar uma instância e a chave privada correspondente sempre que se conectar à instância.

   Warning

   Não selecione a opção Proceed without a key pair. Se você executar sua instância sem um par de chaves, você não poderá conectá-la.

   Quando estiver pronto, escolhaExecutar instâncias.
8. Uma página de confirmação informa que sua instância está sendo executada. Selecione Visualizar instâncias para fechar a página de confirmação e voltar ao console.
9. Na tela Instances, é possível visualizar o status da execução. Demora um pouco para executar uma instância. Ao executar uma instância, seu estado inicial é `pending`. Após a inicialização da instância, seu estado muda para `running` e ela recebe um nome DNS público. (Se a coluna Public DNS (IPv4) estiver oculta, escolha Mostrar/ocultar colunas (o ícone de engrenagem) no canto superior direito da página e selecione Public DNS (IPv4).)
10. Pode levar alguns minutos até que a instância esteja pronta para que você possa se conectar a ela. Certifique-se de que sua instância tenha sido aprovada nas verificações de status. É possível visualizar essas informações na coluna Status Checks.

    Depois que sua nova instância tiver passado nas verificações de status, continue para o próximo procedimento e conecte-se a ela.

Para se conectar à sua instância do

Você pode se conectar a uma instância usando cliente baseado em navegador selecionando a instância no console da Amazon EC2 e optando por se conectar usando o Amazon EC2 Instance Connect. O Instance Connect lida com as permissões e fornece uma conexão bem-sucedida.

1. Abra o console do Amazon EC2 em https://console.aws.amazon.com/ec2/.
2. No menu esquerdo, escolhaInstâncias.
3. Selecione a instância e escolha Connect (Conectar).
4. SelecioneInstance Connect do Amazon EC2 (conexão SSH baseada no navegador),Conecte-se.

Agora, você deve ter umAmazon EC2 Instance ConnectJanela que está conectada à nova instância do Amazon EC2.

Instale o Git, Node.js e configure oAWS CLI

Nesta seção, você instalará o Git e o Node.js, em sua instância do Linux.

Para instalar o Git

1. NoAmazon EC2 Instance ConnectJanela, atualize sua instância usando o seguinte comando.

   ```
   sudo yum update -y
   ```

2. NoAmazon EC2 Instance ConnectJanela, instale o Git usando o seguinte comando.

```
sudo yum install git -y
```

Para instalar o Node.js

1. NoAmazon EC2 Instance Connect, instale o gerenciador de versão do nó (nvm) usando o comando a seguir.

   ```
   curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.34.0/install.sh | bash
   ```

   Usaremos o nvm para instalar o Node.js, pois o nvm pode instalar várias versões do Node.js e permitir que você alterne entre elas.
2. NoAmazon EC2 Instance Connect, ative o nvm usando este comando.

   ```
   . ~/.nvm/nvm.sh
   ```
3. NoAmazon EC2 Instance Connect, use o nvm para instalar a versão mais recente do Node.js usando esse comando.

   ```
   nvm install node
   ```

   Instalar Node.js também instala o gerenciador de pacotes do nó (npm, do inglês "node package manager") para que você possa instalar módulos adicionais, conforme necessário.
4. NoAmazon EC2 Instance Connect, verifique se o Node.js está instalado e funcionando corretamente usando esse comando.

   ```
   node -v
   ```

   Este tutorial requer o Nó v10.0 ou posterior.

Como configurar o AWS CLI

Sua instância do Amazon EC2 vem pré-carregada com aAWS CLI. No entanto, você deve concluir oAWS CLI. Para obter mais informações sobre como configurar sua CLI, consulteConfigurar aAWS CLI.

1. O exemplo a seguir mostra valores de amostra. Substitua-os por seus próprios valores. Você pode encontrar esses valores em seuAWSconsole nas informações da sua conta emCredenciais de segurança.

   NoAmazon EC2 Instance ConnectJanela, insira este comando:

   ```
   aws configure
   ```

   Em seguida, insira os valores da sua conta nos prompts exibidos.

   ```
   AWS Access Key ID [None]: AKIAIOSFODNN7EXAMPLE
   AWS Secret Access Key [None]: wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
   Default region name [None]: us-west-2
   Default output format [None]: json
   ```
2. Você pode testar seuAWS CLIconfiguração com este comando:

```
aws iot describe-endpoint --endpoint-type iot:Data-ATS
```

Se suas receitasAWS CLIestá configurado corretamente, o comando deve retornar um endereço de endpoint do seuConta da AWS.

CriarAWS IoTrecursos para seu dispositivo virtual

Esta seção descreve como usar oAWS CLIpara criar o objeto coisa e seus arquivos de certificado diretamente no dispositivo virtual. Isso é feito diretamente no dispositivo para evitar a possível complicação que pode surgir ao copiá-los para o dispositivo de outro computador.

Para criar umAWS IoTobjeto de coisa na instância do Linux

Dispositivos conectados aoAWS IoTsão representados porobjetos da coisa danoAWS IoTRegistro. UMAthing objectrepresenta um dispositivo específico ou uma entidade lógica. Nesse caso, othing objectRepresentará seu dispositivo virtual, esta instância do Amazon EC2.

1. NoAmazon EC2 Instance Connectjanela, execute o seguinte comando para criar seu objeto da coisa.

```
aws iot create-thing --thing-name "MyIotThing"
```

2. A resposta JSON deve ser semelhante a:

```
{
    "thingArn": "arn:aws:iot:your-region:your-aws-account:thing/MyIotThing",
    "thingName": "MyIotThing",
    "thingId": "6cf922a8-d8ea-4136-f3401EXAMPLE"
}
```

Para criar e anexarAWS IoTchaves e certificados em sua instância do Linux

Ocreate-keys-and-certificateO comando cria certificados de cliente assinados pela autoridade de certificação raiz da Amazon. Esse certificado é usado para autenticar a identidade do dispositivo virtual.

1. NoAmazon EC2 Instance Connect, crie um diretório para armazenar os arquivos de certificado e chave.

```
mkdir ~/certs
```

2. NoAmazon EC2 Instance Connect, baixe uma cópia do certificado da autoridade de certificação (CA) da Amazon usando este comando.

```
curl -o ~/certs/Amazon-root-CA-1.pem \
    https://www.amazontrust.com/repository/AmazonRootCA1.pem
```

3. NoAmazon EC2 Instance Connect, execute o comando a seguir para criar sua chave privada, chave pública e arquivos de certificado X.509. Este comando também registra e ativa o certificado comAWS IoT.

```
aws iot create-keys-and-certificate \
    --set-as-active \
    --certificate-pem-outfile "~/certs/device.pem.crt" \
    --public-key-outfile "~/certs/public.pem.key" \
    --private-key-outfile "~/certs/private.pem.key"
```

A resposta é semelhante à seguinte. Salvar `ocertificateArn`para que você possa usá-lo em comandos subseqüentes. Você precisará dele para anexar seu certificado à sua coisa e anexar a política ao certificado em etapas posteriores.

```
{
    "certificateArn": "arn:aws:iot:us-
west-2:123456789012:cert/9894ba17925e663f1d29c23af4582b8e3b7619c31f3fbd93adcb51ae54b83dc2",
    "certificateId":
 "9894ba17925e663f1d29c23af4582b8e3b7619c31f3fbd93adcb51ae54b83dc2",
    "certificatePem": "
-----BEGIN CERTIFICATE-----
MIICiTCCEXAMPLE6m7oRw0uXOjANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC
VVMxCzAJBgNVBAgEXAMPLEAwDgYDVQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6
b24xFDASBgNVBAsTC0lBTSEXAMPLE2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAd
BgkqhkiG9w0BCQEWEG5vb25lQGFtYEXAMPLEb20wHhcNMTEwNDI1MjA0NTIxWhcN
MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCEXAMPLEJBgNVBAgTAldBMRAwDgYD
VQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6b24xFDAEXAMPLEsTC0lBTSBDb25z
b2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAdBgkqhkiG9w0BCQEXAMPLE25lQGFt
YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+aEXAMPLE
EXAMPLEfEvySWtC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEIO3IyNoH/f0wYK8m9T
rDHudUZEXAMPLELG5M43q7Wgc/MbQITxOUSQv7c7ugFFDzQGBzZswY6786m86gpE
Ibb3OhjZnzcvQAEXAMPLEWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
nUhVVxYUntneD9+h8Mg9qEXAMPLEyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb
FFBjvSfpJIlJ00zbhNYS5f6GuoEDEXAMPLEBHjJnyp378OD8uTs7fLvjx79LjSTb
NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE=
-----END CERTIFICATE-----\n",
    "keyPair": {
        "PublicKey": "-----BEGIN PUBLIC
 KEY-----\nMIIBIjANBgkqhkEXAMPLEQEFAAOCAQ8AMIIBCgKCAQEAEXAMPLE1nnyJwKSMHw4h
\nMMEXAMPLEuuN/dMAS3fyce8DW/4+EXAMPLEyjmoF/YVF/
gHr99VEEXAMPLE5VF13\n59VK7cEXAMPLE67GK+y+jikqXOgHh/xJTwo
+sGpWEXAMPLEDz18xOd2ka4tCzuWEXAMPLEahJbYkCPUBSU8opVkR7qkEXAMPLE1DR6sx2HocliOOLtu6Fkw91swQWEXAMPLE
\GB3ZPrNh0PzQYvjUStZeccyNCx2EXAMPLEvp9mQOUXP6plfgxwKRX2fEXAMPLEDa\nhJLXkX3rHU2xbxJSq7D
+XEXAMPLEcw+LyFhI5mgFRl88eGdsAEXAMPLElnI9EesG\nFQIDAQAB\n-----END PUBLIC KEY-----\n",
        "PrivateKey": "-----BEGIN RSA PRIVATE KEY-----\nkey omitted for security
 reasons\n-----END RSA PRIVATE KEY-----\n"
    }
}
```

4. NoAmazon EC2 Instance Connectjanela, anexe seu objeto coisa ao certificado que você acabou de criar usando o seguinte comando e o*`certificateArn`*na resposta do comando anterior.

```
aws iot attach-thing-principal \
    --thing-name "MyIotThing" \
    --principal "certificateArn"
```

Se houver êxito, o comando não exibirá nenhuma saída.

Para criar e anexar uma política

1. NoAmazon EC2 Instance Connect, crie o arquivo de política copiando e colando este documento de política em um arquivo chamado**`~/policy.json`**.

   Se não tiver um editor Linux favorito, você pode abrirnano, usando este comando.

```
nano ~/policy.json
```

   E colando o documento de política para`policy.json`dentro dele. Digite ctrl-x para sair donanoEditor e salve o arquivo.

Conteúdo do documento de política para`policy.json`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:Publish",
                "iot:Subscribe",
                "iot:Receive",
                "iot:Connect"
            ],
            "Resource": [
                "*"
            ]
        }
    ]
}
```

2. NoAmazon EC2 Instance Connect, crie sua política usando o seguinte comando.

```
aws iot create-policy \
    --policy-name "MyIotThingPolicy" \
    --policy-document "file://~/policy.json"
```

Resultado:

```
{
    "policyName": "MyIotThingPolicy",
    "policyArn": "arn:aws:iot:your-region:your-aws-account:policy/MyIotThingPolicy",
    "policyDocument": "{
        \"Version\": \"2012-10-17\",
        \"Statement\": [
            {
                \"Effect\": \"Allow\",
                \"Action\": [
                    \"iot:Publish\",
                    \"iot:Receive\",
                    \"iot:Subscribe\",
                    \"iot:Connect\"
                ],
                \"Resource\": [
                    \"*\"
                ]
            }
        ]
    }",
    "policyVersionId": "1"
}
```

3. NoAmazon EC2 Instance Connect, anexe a política ao certificado do dispositivo virtual usando o comando a seguir.

```
aws iot attach-policy \
    --policy-name "MyIotThingPolicy" \
    --target "certificateArn"
```

Se houver êxito, o comando não exibirá nenhuma saída.

Neste ponto, você criou para seu dispositivo virtual:

- Um objeto coisa para representar seu dispositivo virtual emAWS IoT.
- Um certificado para autenticar seu dispositivo virtual.
- Um documento de política para autorizar seu dispositivo virtual a Connect aoAWS IoTe para publicar, receber e assinar mensagens.

Instalar o SDK de dispositivos da AWS IoT para JavaScript

Nesta seção, você instalará oAWS IoTSDK do dispositivo para JavaScript, que contém o código que os aplicativos podem usar para se comunicar comAWS IoTe os programas de amostra.

Para instalar oAWS IoTSDK de dispositivo da para JavaScript na instância do Linux

1. NoAmazon EC2 Instance ConnectJanela, clone oAWS IoTSDK de dispositivo da para JavaScript repositório no`aws-iot-device-sdk-js-v2`diretório do seu diretório inicial usando este comando.

```
cd ~
git clone https://github.com/aws/aws-iot-device-sdk-js-v2.git
```

2. Navegue até o `.aws-iot-device-sdk-js-v2`Diretório que você criou na etapa anterior.

```
cd aws-iot-device-sdk-js-v2
```

3. Use o npm para instalar o SDK.

```
npm install
```

Execute o aplicativo de amostra

Os comandos nas próximas seções pressupõem que seus arquivos de chave e certificado estão armazenados no dispositivo virtual, conforme mostrado nesta tabela.

Nomes de arquivos de certificado

Arquivo	Caminho do arquivo
Chave privada	`~/certs/private.pem.key`
Certificado de dispositivo	`~/certs/device.pem.crt`
Certificado CA raiz	`~/certs/Amazon-root-CA-1.pem`

Nesta seção, você instalará e executará `opub-sub.js`aplicativo de amostra encontrado no`aws-iot-device-sdk-js-v2/samples/node`Diretório doAWS IoTSDK de dispositivo da para JavaScript. Este aplicativo mostra como um dispositivo, sua instância do Amazon EC2, usa a biblioteca MQTT para publicar e assinar mensagens MQTT. O`pub-sub.js`aplicativo de amostra se inscreve em um tópico,`topic_1`, publica 10 mensagens nesse tópico e exibe as mensagens à medida que são recebidas do corretor de mensagens.

Para instalar e executar o aplicativo de amostra

1. NoAmazon EC2 Instance Connect, navegue até a janela`aws-iot-device-sdk-js-v2/samples/node/pub_sub`diretório que o SDK criou e instalou o aplicativo de amostra usando esses comandos.

```
cd ~/aws-iot-device-sdk-js-v2/samples/node/pub_sub
npm install
```

2. NoAmazon EC2 Instance Connectjanela, get*seu ponto de extremidade iot-*desdeAWS IoTusando este comando.

```
aws iot describe-endpoint --endpoint-type iot:Data-ATS
```

3. NoAmazon EC2 Instance Connectjanela, insira*seu ponto de extremidade iot-*conforme indicado e execute este comando.

```
node dist/index.js --topic topic_1 --ca_file ~/certs/Amazon-root-CA-1.pem --cert ~/
certs/device.pem.crt --key ~/certs/private.pem.key --endpoint your-iot-endpoint
```

O aplicativo de exemplo:

1. Conecta-se aoAWS IoTserviço para sua conta.
2. Assina o tópico da mensagem,topic_1e exibe as mensagens que ele recebe sobre esse tópico.
3. Publica 10 mensagens para o tópico,topic_1.
4. Exibe uma saída semelhante à seguinte:

```
Publish received on topic topic_1
{"message":"Hello world!","sequence":1}
Publish received on topic topic_1
{"message":"Hello world!","sequence":2}
Publish received on topic topic_1
{"message":"Hello world!","sequence":3}
Publish received on topic topic_1
{"message":"Hello world!","sequence":4}
Publish received on topic topic_1
{"message":"Hello world!","sequence":5}
Publish received on topic topic_1
{"message":"Hello world!","sequence":6}
Publish received on topic topic_1
{"message":"Hello world!","sequence":7}
Publish received on topic topic_1
{"message":"Hello world!","sequence":8}
Publish received on topic topic_1
{"message":"Hello world!","sequence":9}
Publish received on topic topic_1
{"message":"Hello world!","sequence":10}
```

Se estiver tendo problemas para executar o aplicativo de exemplo, consultethe section called “Solução de problemas com o aplicativo de amostra” (p. 65).

Você também pode adicionar o`--verbosity Debug`parâmetro para a linha de comando para que o aplicativo de amostra exiba mensagens detalhadas sobre o que ele está fazendo. Essas informações podem fornecer a ajuda necessária para corrigir o problema.

Exibir mensagens do aplicativo de amostra naAWS IoTconsole

Você pode ver as mensagens do aplicativo de amostra à medida que elas passam pelo corretor de mensagens usando oCliente MQTTnoAWS IoTconsole.

Para exibir as mensagens MQTT publicadas pelo aplicativo de amostra

1. Analise Visualizar mensagens MQTT com o cliente MQTT do AWS IoT (p. 66). Isso ajuda a aprender a usar oCliente MQTTnoAWS IoTconsoleComo visualizar mensagens MQTT à medida que passam pelo corretor de mensagens.
2. Abrir oCliente MQTTnoAWS IoTconsole.
3. Inscrever-se no tópico dotopic_1.
4. NoAmazon EC2 Instance Connect, execute o aplicativo de amostra novamente e assista às mensagens noCliente MQTTnoAWS IoTconsole.

```
cd ~/aws-iot-device-sdk-js-v2/samples/node/pub_sub
node dist/index.js --topic topic_1 --ca_file ~/certs/Amazon-root-CA-1.pem --cert ~/
certs/device.pem.crt --key ~/certs/private.pem.key --endpoint your-iot-endpoint
```

Use seu PC ou Mac Windows ou Linux como umAWS IoTdispositivo

Neste tutorial, você configurará um computador pessoal para uso comAWS IoT. Essas instruções são compatíveis com PCs e Macs Windows e Linux. Para fazer isso, instale algum software no computador. Se você não quiser instalar software em seu computador, você pode tentarCriar um dispositivo virtual com o Amazon EC2 (p. 43), que instala todo o software em uma máquina virtual.

Neste tutorial, você irá:

- Configurar seu computador pessoal (p. 51)
- Instalar o Git, o Python e oAWS IoTSDK de dispositivo da para Python (p. 51)
- Configurar a política e execute o aplicativo de exemplo (p. 54)
- Exibir mensagens do aplicativo de amostra naAWS IoTconsole (p. 57)

Configurar seu computador pessoal

Para concluir este tutorial, você precisa de um PC Windows ou Linux ou um Mac com conexão com a Internet.

Antes de prosseguir para a etapa seguinte, verifique se você pode abrir uma janela de linha de comando no computador. Usar ocmd.exeem um PC com Windows. Em um PC Linux ou Mac, useTerminal.

Instalar o Git, o Python e oAWS IoTSDK de dispositivo da para Python

Nesta seção, você instalará o Python e oAWS IoTSDK de dispositivo da para Python no computador.

Instalar a versão mais recente do Git e Python

Para baixar e instalar o Git e o Python no seu computador

1. Verifique se o Git está instalado no computador. Insira este comando na linha de comando.

   ```
   git --version
   ```

   Se o comando exibir a versão do Git, o Git será instalado e você poderá continuar para a próxima etapa.

   Se o comando exibir um erro, abrahttps://git-scm.com/downloade instale o Git para seu computador.

2. Verifique se você já instalou o Python. Insira este comando na linha de comando.

```
python -V
```

> **Note**
>
> Se esse comando der um erro:`Python was not found`, pode ser porque seu sistema operacional chama o executável Python v3.x como`Python3`. Nesse caso, substitua todas as instâncias do`python`com`python3`E continue o restante deste tutorial.

Se o comando exibir a versão do Python, o Python já está instalado. Este tutorial requer Python v3.5 ou posterior.

3. Se o Python estiver instalado, você poderá ignorar o restante das etapas desta seção. Caso contrário, continue.
4. Aberto https://www.python.org/downloads/ e baixe o instalador do seu computador.
5. Se o download não começar a instalar automaticamente, execute o programa baixado para instalar o Python.
6. Verifique a instalação do Python.

```
python -V
```

Confirme se o comando exibe a versão do Python. Se a versão do Python não for exibida, tente baixar e instalar o Python novamente.

Instalar aAWS IoTSDK de dispositivo da para Python

Para instalar oAWS IoTSDK de dispositivo da para Python no computador

1. Instale v2 doAWS IoTSDK de dispositivo para Python.

```
python3 -m pip install awsiotsdk
```

2. Clone oAWS IoTSDK de dispositivo do repositório do Python no diretório aws-iot-device-sdk-python-v2 do diretório inicial. Este procedimento se refere ao diretório base para os arquivos que você está instalando como*casa*.

   A localização real do*casa*O diretório depende do sistema operacional.

   Linux/macOS

   No macOS e Linux, a ação*casa*directory é~.

```
cd ~
git clone https://github.com/aws/aws-iot-device-sdk-python-v2.git
```

   Windows

   No Windows, você pode encontrar o*casa*caminho de diretório executando este comando na`cmd`.

```
echo %USERPROFILE%
cd %USERPROFILE%
git clone https://github.com/aws/aws-iot-device-sdk-python-v2.git
```

Note

Se você estiver usando o Windows PowerShell em oposição acmd.exeDepois use o comando a seguir.

```
echo $home
```

Preparar-se para executar os aplicativos de exemplo

Para preparar o sistema para executar o aplicativo de exemplo

- Criação do`certs`. Para o`certs`diretório, copie os arquivos de chave privada, certificado de dispositivo e certificado CA raiz salvos quando criou e registrou o objeto de coisa nothe section called “CriarAWS IoTrecursos” (p. 39). Os nomes de arquivo de cada arquivo no diretório de destino devem corresponder aos da tabela.

  Os comandos na próxima seção pressupõem que seus arquivos de chave e certificado estão armazenados no dispositivo, conforme mostrado nesta tabela.

  Linux/macOS

  Execute este comando para criar o`certs`Subdiretório que você usará ao executar os aplicativos de exemplo.

  ```
  mkdir ~/certs
  ```

  No novo subdiretório, copie os arquivos para os caminhos do arquivo de destino mostrados na tabela a seguir.

  Nomes de arquivos de certificado

Arquivo	Caminho do arquivo
Chave privada	`~/certs/private.pem.key`
Certificado de dispositivo	`~/certs/device.pem.crt`
Certificado CA raiz	`~/certs/Amazon-root-CA-1.pem`

  Execute este comando para listar os arquivos na`certs`diretório e compare-os com os listados na tabela.

  ```
  ls -l ~/certs
  ```

  Windows

  Execute este comando para criar o`certs`Subdiretório que você usará ao executar os aplicativos de exemplo.

  ```
  mkdir %USERPROFILE%\certs
  ```

No novo subdiretório, copie os arquivos para os caminhos do arquivo de destino mostrados na tabela a seguir.

Nomes de arquivos de certificado

Arquivo	Caminho do arquivo
Chave privada	`%USERPROFILE%\certs\private.pem.key`
Certificado de dispositivo	`%USERPROFILE%\certs\device.pem.crt`
Certificado CA raiz	`%USERPROFILE%\certs\Amazon-root-CA-1.pem`

Execute este comando para listar os arquivos na`certs`diretório e compare-os com os listados na tabela.

```
dir %USERPROFILE%\certs
```

Configurar a política e execute o aplicativo de exemplo

Nesta seção, você configurará sua política e executará o`pubsub.py`script de amostra encontrado na`aws-iot-device-sdk-python-v2/samples`Diretório doAWS IoT Device SDK for Python. Esse script mostra como o dispositivo usa a biblioteca MQTT para publicar e assinar mensagens MQTT.

O`pubsub.py`aplicativo de amostra se inscreve em um tópico,`test/topic`, publica 10 mensagens nesse tópico e exibe as mensagens à medida que são recebidas do corretor de mensagens.

Para executar a`pubsub.py`Script de exemplo, você precisa das seguintes informações:

Direcionar valores de parâmetro

Parâmetro	Onde encontrar o valor
`seu ponto de extremidade iot-`	1. NoAWS IoTconsoleNo menu esquerdo, escolhaConfigurações. 2. NoConfigurações, seu endpoint é exibido naPonto de extremidade de dados de dispositivoseção.

O*`seu ponto de extremidade iot-`*value tem um formato de:*`endpoint_id`*`-ats.iot.`*`region`*`.amazonaws.com`, por exemplo,`a3qj468EXAMPLE-ats.iot.us-west-2.amazonaws.com`.

Antes de executar o script, verifique se a política da sua coisa fornece permissões para que o script de exemplo se conecte, assine, publique e receba.

Para localizar e revisar o documento de política para um recurso coisa

1. NoAWSconsole, noCoisaslist, encontre o recurso coisa que representa seu dispositivo.
2. Selecione oName (Nome)link do recurso coisa que representa seu dispositivo para abrir oDetalhes da coisa.

3. NoDetalhes da coisaA página, noCertificados, escolha o certificado que está anexado ao recurso coisa. Só deve haver um certificado na lista. Se houver mais de um, escolha o certificado cujos arquivos estão instalados no seu dispositivo e que será usado para se conectar aoAWS IoT.
4. NoCertificadopágina de detalhes, naPolíticas, escolha a política anexada ao certificado. Deve haver apenas um. Se houver mais de um, repita a próxima etapa para cada um para garantir que pelo menos uma política conceda o acesso necessário.
5. NoPolíticapágina de visão geral, encontre o editor JSON e escolhaEditar documento de políticapara revisar e editar o documento de política conforme necessário.
6. A política JSON é exibida no exemplo a seguir. No`"Resource"`elemento, substitua*`region:account`*Com suas despesasRegião da AWSeConta da AWSem cada um dos`Resource`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:Publish",
                "iot:Receive"
            ],
            "Resource": [
                "arn:aws:iot:region:account:topic/test/topic"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iot:Subscribe"
            ],
            "Resource": [
                "arn:aws:iot:region:account:topicfilter/test/topic"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iot:Connect"
            ],
            "Resource": [
                "arn:aws:iot:region:account:client/test-*"
            ]
        }
    ]
}
```

Linux/macOS

Para executar o script de amostra no Linux/macOS

1. Na janela da linha de comando, navegue até o`~/aws-iot-device-sdk-python-v2/samples/node/pub_sub`diretório que o SDK criou usando esses comandos.

```
cd ~/aws-iot-device-sdk-python-v2/samples
```

2. Na janela da linha de comando, substitua*`seu ponto de extremidade iot-`*conforme indicado e execute este comando.

```
python3 pubsub.py --endpoint your-iot-endpoint --ca_file ~/certs/Amazon-root-
CA-1.pem --cert ~/certs/device.pem.crt --key ~/certs/private.pem.key
```

Windows

Para executar o aplicativo de exemplo em um PC com Windows

1. Na janela da linha de comando, navegue até o`%USERPROFILE%\aws-iot-device-sdk-python-v2\samples`diretório que o SDK criou e instalou o aplicativo de amostra usando esses comandos.

```
cd %USERPROFILE%\aws-iot-device-sdk-python-v2\samples
```

2. Na janela da linha de comando, substitua*`seu ponto de extremidade iot`*-conforme indicado e execute este comando.

```
python3 pubsub.py --endpoint your-iot-endpoint --ca_file %USERPROFILE%\certs
\Amazon-root-CA-1.pem --cert %USERPROFILE%\certs\device.pem.crt --key %USERPROFILE%
\certs\private.pem.key
```

Exemplo de script:

1. Conecta-se aoAWS IoTserviço para sua conta.
2. Assina o tópico da mensagem,teste/tópicoe exibe as mensagens que ele recebe sobre esse tópico.
3. Publica 10 mensagens para o tópico,teste/tópico.
4. Exibe uma saída semelhante à seguinte:

```
Publish received on topic test/topic
{"message":"Hello world!","sequence":1}
Publish received on topic test/topic
{"message":"Hello world!","sequence":2}
Publish received on topic test/topic
{"message":"Hello world!","sequence":3}
Publish received on topic test/topic
{"message":"Hello world!","sequence":4}
Publish received on topic test/topic
{"message":"Hello world!","sequence":5}
Publish received on topic test/topic
{"message":"Hello world!","sequence":6}
Publish received on topic test/topic
{"message":"Hello world!","sequence":7}
Publish received on topic test/topic
{"message":"Hello world!","sequence":8}
Publish received on topic test/topic
{"message":"Hello world!","sequence":9}
Publish received on topic test/topic
{"message":"Hello world!","sequence":10}
```

Se estiver tendo problemas para executar o aplicativo de exemplo, consultethe section called “Solução de problemas com o aplicativo de amostra” (p. 65).

Você também pode adicionar o`--verbosity Debug`parâmetro para a linha de comando para que o aplicativo de amostra exiba mensagens detalhadas sobre o que ele está fazendo. Essas informações podem ajudá-lo a corrigir o problema.

Exibir mensagens do aplicativo de amostra naAWS IoTconsole

Você pode ver as mensagens do aplicativo de amostra à medida que elas passam pelo corretor de mensagens usando oCliente MQTTnoAWS IoTconsole.

Para exibir as mensagens MQTT publicadas pelo aplicativo de amostra

1. Analise Visualizar mensagens MQTT com o cliente MQTT do AWS IoT (p. 66). Isso ajuda a aprender a usar oCliente MQTTnoAWS IoTconsolePara ver mensagens MQTT à medida que passam pelo operador de mensagens.
2. Abrir oCliente MQTTnoAWS IoTconsole.
3. Inscrever-se no tópico doteste/tópico.
4. Na janela da linha de comando, execute o aplicativo de exemplo novamente e observe as mensagens naCliente MQTTnoAWS IoTconsole.

   Linux/macOS

   ```
   cd ~/aws-iot-device-sdk-python-v2/samples
   python3 pubsub.py --topic test/topic --ca_file ~/certs/Amazon-root-CA-1.pem --cert
    ~/certs/device.pem.crt --key ~/certs/private.pem.key --endpoint your-iot-endpoint
   ```

   Windows

   ```
   cd %USERPROFILE%\aws-iot-device-sdk-python-v2\samples
   python3 pubsub.py --topic test/topic --ca_file %USERPROFILE%\certs\Amazon-root-
   CA-1.pem --cert %USERPROFILE%\certs\device.pem.crt --key %USERPROFILE%\certs
   \private.pem.key --endpoint your-iot-endpoint
   ```

Connect um Raspberry Pi ou outro dispositivo

Nesta seção, configuraremos um Raspberry Pi para uso comAWS IoT. Se você tiver outro dispositivo que gostaria de conectar, as instruções para o Raspberry Pi incluem referências que podem ajudá-lo a adaptar essas instruções ao seu dispositivo.

Isso normalmente leva cerca de 20 minutos, mas pode demorar mais se você tiver muitas atualizações de software do sistema para instalar.

Neste tutorial, você irá:

- Configurar o dispositivo (p. 58)
- Instale as ferramentas e bibliotecas necessárias para oAWS IoTSDK de dispositivo da (p. 58)
- InstallAWS IoTSDK de dispositivo da (p. 59)
- Instalar e executar o aplicativo de amostra (p. 61)
- Exibir mensagens do aplicativo de amostra naAWS IoTconsole (p. 64)

> **Important**
>
> Adaptar essas instruções a outros dispositivos e sistemas operacionais pode ser um desafio. Você precisará entender o dispositivo bem o suficiente para poder interpretar essas instruções e aplicá-las ao seu dispositivo.
> Se você encontrar dificuldades ao configurar seu dispositivo paraAWS IoT, não podemos oferecer assistência além das instruções desta seção. No entanto, você pode tentar uma das outras opções de dispositivo como alternativa, comoCriar um dispositivo virtual com o Amazon EC2 (p. 43)ouUse seu PC ou Mac Windows ou Linux como umAWS IoTdispositivo (p. 51).

Configurar o dispositivo

O objetivo desta etapa é coletar o que você precisará para configurar seu dispositivo de forma que ele possa iniciar o sistema operacional (SO), conectar-se à Internet e permitir que você interaja com ele em uma interface de linha de comando.

Para concluir este tutorial, você precisará do seguinte:

- Uma Conta da AWS. Se você ainda não tiver um, conclua as etapas descritas emConfigurar seuConta da AWS (p. 19)Antes de continuar.
- UMARaspberry Pi 3 Modelo Bou modelo mais recente. Isso pode funcionar em versões anteriores do Raspberry Pi, mas elas não foram testadas.
- Raspberry Pi OS (32 bits)ou posterior. Recomendamos o uso da versão mais recente do sistema operacional Raspberry Pi. Versões anteriores do sistema operacional podem funcionar, mas elas não foram testadas.

  Para executar este exemplo, você não precisa instalar a área de trabalho com a interface gráfica do usuário (GUI); no entanto, se você é novo no Raspberry Pi e seu hardware Raspberry Pi suportar, usar a área de trabalho com a GUI pode ser mais fácil.
- Uma conexão Ethernet ou Wi-Fi.
- Teclado, mouse, monitor, cabos, fontes de alimentação e outros hardwares exigidos pelo seu dispositivo.

> **Important**
>
> Antes de prosseguir para a próxima etapa, seu dispositivo deve ter seu sistema operacional instalado, configurado e em execução. O dispositivo deve estar conectado à Internet e você precisará acessar o dispositivo usando sua interface de linha de comando. O acesso à linha de comando pode ser através de um teclado, mouse e monitor conectados diretamente ou usando uma interface remota de terminal SSH.

Se você estiver executando um sistema operacional em seu Raspberry Pi que tenha uma interface gráfica do usuário (GUI), abra uma janela de terminal no dispositivo e execute as instruções a seguir nessa janela. Caso contrário, se você estiver se conectando ao seu dispositivo usando um terminal remoto, como PuTTY, abra um terminal remoto para o dispositivo e use-o.

Instale as ferramentas e bibliotecas necessárias para oAWS IoTSDK de dispositivo da

Antes de instalar oAWS IoTSDK do dispositivo e código de exemplo, certifique-se de que seu sistema seja up-to-date e tem as ferramentas e bibliotecas necessárias para instalar os SDKs.

1. Atualize o sistema operacional e instale as bibliotecas necessárias

   Antes de instalar umAWS IoTSDK de dispositivo, execute esses comandos em uma janela de terminal no dispositivo para atualizar o sistema operacional e instalar as bibliotecas necessárias.

   ```
   sudo apt-get update
   ```

   ```
   sudo apt-get upgrade
   ```

   ```
   sudo apt-get install cmake
   ```

   ```
   sudo apt-get install libssl-dev
   ```

2. Instalação do Git

Se o sistema operacional do seu dispositivo não vier com o Git instalado, você precisará instalá-lo para instalar oAWS IoTSDK de dispositivo da para JavaScript.

a. Verifique se o Git já está instalado executando esse comando.

```
git --version
```

b. Se o comando anterior retornar a versão do Git, o Git já está instalado e você poderá pular para a Etapa 3.

c. Se um erro for exibido quando você executar ogit, instale o Git executando este comando.

```
sudo apt-get install git
```

d. Teste novamente para ver se o Git está instalado executando este comando.

```
git --version
```

e. Se o Git estiver instalado, continue para a próxima seção. Caso contrário, solucione e corrija o erro antes de continuar. Você precisa do Git para instalar oAWS IoTSDK de dispositivo da para JavaScript.

InstallAWS IoTSDK de dispositivo da

Instalar aAWS IoTSDK de dispositivo da.

Python

Nesta seção, você instalará o Python, suas ferramentas de desenvolvimento e oAWS IoTSDK de dispositivo para Python em seu dispositivo. Essas instruções são para um Raspberry Pi executando o mais recente sistema operacional Raspberry Pi. Se você tiver outro dispositivo ou estiver usando outro sistema operacional, poderá ser necessário adaptar essas instruções para seu dispositivo.

1. Instalar o Python e suas ferramentas de desenvolvimento

   OAWS IoTO SDK do dispositivo para Python requer que o Python v3.5 ou posterior seja instalado em seu Raspberry Pi.

   Em uma janela de terminal do seu dispositivo, execute esses comandos.

   1. Execute esse comando para determinar a versão do Python instalada no dispositivo.

      ```
      python3 --version
      ```

      Se o Python estiver instalado, ele exibirá sua versão.
   2. Se a versão exibida for`Python 3.5`Ou superior, pule para a Etapa 2.
   3. Se a versão exibida for menor que`Python 3.5`, você pode instalar a versão correta executando este comando.

      ```
      sudo apt install python3
      ```
   4. Execute este comando para confirmar que a versão correta do Python agora está instalada.

      ```
      python3 --version
      ```
2. Teste para pip3

Em uma janela de terminal do seu dispositivo, execute esses comandos.

1. Execute este comando para ver sepip3está instalado.

```
pip3 --version
```

2. Se o comando retornar um número de versão,pip3está instalado e você pode pular para a Etapa 3.
3. Se o comando anterior retornar um erro, execute esse comando para instalarpip3.

```
sudo apt install python3-pip
```

4. Execute este comando para ver sepip3está instalado.

```
pip3 --version
```

3. Instalar o atualAWS IoTSDK de dispositivo da para Python

   Instalar aAWS IoTSDK do dispositivo para Python e baixe os aplicativos de amostra para o seu dispositivo.

   No dispositivo, execute esses comandos.

```
cd ~
python3 -m pip install awsiotsdk
```

```
git clone https://github.com/aws/aws-iot-device-sdk-python-v2.git
```

JavaScript

Nesta seção, você instalará o Node.js, o gerenciador de pacotes npm e oAWS IoTSDK de dispositivo da para JavaScript no dispositivo. Essas instruções são para um Raspberry Pi executando o sistema operacional Raspberry Pi. Se você tiver outro dispositivo ou estiver usando outro sistema operacional, poderá ser necessário adaptar essas instruções para seu dispositivo.

1. Instalar a versão mais recente do Node.js

   OAWS IoTSDK de dispositivo da para JavaScript Requer que o Node.js e o gerenciador de pacotes npm sejam instalados no Raspberry Pi.

   a. Faça download da versão mais recente do repositório Node inserindo esse comando.

```
cd ~
curl -sL https://deb.nodesource.com/setup_12.x | sudo -E bash -
```

   b. Instale o Node e o npm.

```
sudo apt-get install -y nodejs
```

   c. Verifique a instalação do Node.

```
node -v
```

Confirme se o comando exibe a versão do nó. Este tutorial requer o Nó v10.0 ou posterior. Se a versão do Node não for exibida, tente baixar o repositório Node novamente.

d. Verifique a instalação do npm.

```
npm -v
```

Confirme se o comando exibe a versão npm. Se a versão npm não for exibida, tente instalar o Node e o npm novamente.

e. Reinicie o dispositivo.

```
sudo shutdown -r 0
```

Continue depois que o dispositivo for reiniciado.

2. Instalar o SDK de dispositivos da AWS IoT para JavaScript

Instalar aAWS IoTSDK de dispositivo da para JavaScript No Raspberry Pi.

a. Clone oAWS IoTSDK de dispositivo da para JavaScript repositório no`aws-iot-device-sdk-js-v2`Diretório do*casa*. No Raspberry Pi, o*casa*directory é`~/`, que é usado como o*casa*diretório nos seguintes comandos. Se o dispositivo usar um caminho diferente para o*casa*diretório, você deve substituir`~/`com o caminho correto para o dispositivo nos seguintes comandos.

Esses comandos criam o`~/aws-iot-device-sdk-js-v2`e copie o código SDK para ele.

```
cd ~
git clone https://github.com/aws/aws-iot-device-sdk-js-v2.git
```

b. Altere para o`aws-iot-device-sdk-js-v2`Diretório que você criou na etapa anterior e executou`npm install`Para instalar o SDK. O comando`npm install`Invocará o`aws-crt`Criar biblioteca que pode levar alguns minutos para ser concluída.

```
cd ~/aws-iot-device-sdk-js-v2
npm install
```

Instalar e executar o aplicativo de amostra

Nesta seção, você instalará e executará o`pubsub`aplicativo de amostra encontrado noAWS IoTSDK de dispositivo da. Este aplicativo mostra como o dispositivo usa a biblioteca MQTT para publicar e assinar mensagens MQTT. O aplicativo de amostra se inscreve em um tópico,`topic_1`, publica 10 mensagens nesse tópico e exibe as mensagens à medida que são recebidas do corretor de mensagens.

Instalar os arquivos de certificado

O aplicativo de amostra requer que os arquivos de certificado que autenticam o dispositivo a ser instalado no dispositivo.

Para instalar os arquivos de certificado do dispositivo para o aplicativo de amostra

1. Criar um`certs`subdiretório em seu*casa*diretório executando esses comandos.

```
cd ~
mkdir certs
```

2. Para o`~/certs`diretório, copie a chave privada, certificado de dispositivo e certificado CA raiz que você criou anteriormente nothe section called “CriarAWS IoTrecursos” (p. 39).

   A forma como você copia os arquivos de certificado para o dispositivo depende do dispositivo e do sistema operacional e não está descrita aqui. No entanto, se o dispositivo suportar uma interface gráfica do usuário (GUI) e tiver um navegador da Web, você poderá executar o procedimento descrito emthe section called “CriarAWS IoTrecursos” (p. 39)no navegador da Web do seu dispositivo para baixar os arquivos resultantes diretamente para o dispositivo.

   Os comandos na próxima seção pressupõem que seus arquivos de chave e certificado estão armazenados no dispositivo, conforme mostrado nesta tabela.

   Nomes de arquivos de certificado

Arquivo	Caminho do arquivo
Certificado CA raiz	`~/certs/Amazon-root-CA-1.pem`
Certificado de dispositivo	`~/certs/certificate.pem.crt`
Chave privada	`~/certs/private.pem.key`

Para executar o aplicativo de exemplo, você precisa das seguintes informações:

Direcionar valores de parâmetro

Parâmetro	Onde encontrar o valor
seu ponto de extremidade iot-	NoAWS IoTconsole, escolhaGerenciee, depois, escolhaCoisas. Escolha a coisa de IoT que você criou para seu dispositivo,MyIotthingFoi o nome usado anteriormente e, em seguida, escolhaInteragir. Na página de detalhes da coisa, seu endpoint é exibido naHTTPSseção. Se você usar o novoAWS IoTConsole do, escolhaConfiguraçõesdoAWS IoTmenu. Seu endpoint é exibido naPonto de extremidade de dados de dispositivoseção.

O*seu ponto de extremidade iot-*value tem um formato de:*endpoint_id*-`ats.iot.`*region*`.amazonaws.com`, por exemplo,`a3qj468EXAMPLE-ats.iot.us-west-2.amazonaws.com`.

Python

Para instalar e executar o aplicativo de amostra

1. Navegue até o diretório de aplicativos de exemplo.

```
cd ~/aws-iot-device-sdk-python-v2/samples
```

2. Na janela da linha de comando, substitua*seu ponto de extremidade iot-*conforme indicado e execute este comando.

```
python3 pubsub.py --topic topic_1 --ca_file ~/certs/Amazon-root-CA-1.pem --cert
 ~/certs/certificate.pem.crt --key ~/certs/private.pem.key --endpoint your-iot-
endpoint
```

3. Observe que o aplicativo de amostra:

   1. Conecta-se aoAWS IoTserviço para sua conta.
   2. Assina o tópico da mensagem,topic_1e exibe as mensagens que ele recebe sobre esse tópico.
   3. Publica 10 mensagens para o tópico,topic_1.
   4. Exibe uma saída semelhante à seguinte:

```
Connecting to a3qEXAMPLEffp-ats.iot.us-west-2.amazonaws.com with client ID
 'test-0c8ae2ff-cc87-49d2-a82a-ae7ba1d0ca5a'...
Connected!
Subscribing to topic 'topic_1'...
Subscribed with QoS.AT_LEAST_ONCE
Sending 10 message(s)
Publishing message to topic 'topic_1': Hello World! [1]
Received message from topic 'topic_1': b'Hello World! [1]'
Publishing message to topic 'topic_1': Hello World! [2]
Received message from topic 'topic_1': b'Hello World! [2]'
Publishing message to topic 'topic_1': Hello World! [3]
Received message from topic 'topic_1': b'Hello World! [3]'
Publishing message to topic 'topic_1': Hello World! [4]
Received message from topic 'topic_1': b'Hello World! [4]'
Publishing message to topic 'topic_1': Hello World! [5]
Received message from topic 'topic_1': b'Hello World! [5]'
Publishing message to topic 'topic_1': Hello World! [6]
Received message from topic 'topic_1': b'Hello World! [6]'
Publishing message to topic 'topic_1': Hello World! [7]
Received message from topic 'topic_1': b'Hello World! [7]'
Publishing message to topic 'topic_1': Hello World! [8]
Received message from topic 'topic_1': b'Hello World! [8]'
Publishing message to topic 'topic_1': Hello World! [9]
Received message from topic 'topic_1': b'Hello World! [9]'
Publishing message to topic 'topic_1': Hello World! [10]
Received message from topic 'topic_1': b'Hello World! [10]'
10 message(s) received.
Disconnecting...
Disconnected!
```

Se estiver tendo problemas para executar o aplicativo de exemplo, consultethe section called “Solução de problemas com o aplicativo de amostra” (p. 65).

Você também pode adicionar o`--verbosity Debug`parâmetro para a linha de comando para que o aplicativo de amostra exiba mensagens detalhadas sobre o que ele está fazendo. Essas informações podem fornecer a ajuda necessária para corrigir o problema.

JavaScript

Para instalar e executar o aplicativo de amostra

1. Na janela da linha de comando, navegue até o`~/aws-iot-device-sdk-js-v2/samples/node/pub_sub`diretório que o SDK criou e instalou o aplicativo de amostra usando esses comandos. O comando`npm install`Invocará o`aws-crt`Criar biblioteca que pode levar alguns minutos para ser concluída.

```
cd ~/aws-iot-device-sdk-js-v2/samples/node/pub_sub
npm install
```

2. Na janela da linha de comando, substitua*seu ponto de extremidade iot-*conforme indicado e execute este comando.

```
node dist/index.js --topic topic_1 --ca_file ~/certs/Amazon-root-CA-1.pem --cert
 ~/certs/certificate.pem.crt --key ~/certs/private.pem.key --endpoint your-iot-
endpoint
```

3. Observe que o aplicativo de amostra:

   1. Conecta-se aoAWS IoTserviço para sua conta.
   2. Assina o tópico da mensagem,topic_1e exibe as mensagens que ele recebe sobre esse tópico.
   3. Publica 10 mensagens para o tópico,topic_1.
   4. Exibe uma saída semelhante à seguinte:

```
Publish received on topic topic_1
{"message":"Hello world!","sequence":1}
Publish received on topic topic_1
{"message":"Hello world!","sequence":2}
Publish received on topic topic_1
{"message":"Hello world!","sequence":3}
Publish received on topic topic_1
{"message":"Hello world!","sequence":4}
Publish received on topic topic_1
{"message":"Hello world!","sequence":5}
Publish received on topic topic_1
{"message":"Hello world!","sequence":6}
Publish received on topic topic_1
{"message":"Hello world!","sequence":7}
Publish received on topic topic_1
{"message":"Hello world!","sequence":8}
Publish received on topic topic_1
{"message":"Hello world!","sequence":9}
Publish received on topic topic_1
{"message":"Hello world!","sequence":10}
```

   Se estiver tendo problemas para executar o aplicativo de exemplo, consultethe section called “Solução de problemas com o aplicativo de amostra” (p. 65).

   Você também pode adicionar o`--verbosity Debug`parâmetro para a linha de comando para que o aplicativo de amostra exiba mensagens detalhadas sobre o que ele está fazendo. Essas informações podem fornecer a ajuda necessária para corrigir o problema.

Exibir mensagens do aplicativo de amostra naAWS IoTconsole

Você pode ver as mensagens do aplicativo de amostra à medida que elas passam pelo corretor de mensagens usando oCliente MQTTnoAWS IoTconsole.

Para exibir as mensagens MQTT publicadas pelo aplicativo de amostra

1. Analise Visualizar mensagens MQTT com o cliente MQTT do AWS IoT (p. 66). Isso ajuda a aprender a usar oCliente MQTTnoAWS IoTconsolePara ver mensagens MQTT à medida que passam pelo operador de mensagens.
2. Abrir oCliente MQTTnoAWS IoTconsole.

3. Inscrever-se no tópico dotopic_1.
4. Na janela da linha de comando, execute o aplicativo de exemplo novamente e observe as mensagens naCliente MQTTnoAWS IoTconsole.

   Python

   ```
   cd ~/aws-iot-device-sdk-python-v2/samples
   python3 pubsub.py --topic topic_1 --ca_file ~/certs/Amazon-root-CA-1.pem --cert
    ~/certs/certificate.pem.crt --key ~/certs/private.pem.key --endpoint your-iot-
   endpoint
   ```

   JavaScript

   ```
   cd ~/aws-iot-device-sdk-js-v2/samples/node/pub_sub
   node dist/index.js --topic topic_1 --ca_file ~/certs/Amazon-root-CA-1.pem --cert
    ~/certs/certificate.pem.crt --key ~/certs/private.pem.key --endpoint your-iot-
   endpoint
   ```

Solução de problemas com o aplicativo de amostra

Se você encontrar um erro ao tentar executar o aplicativo de amostra, aqui estão algumas coisas a serem verificadas.

Verifique o certificado

Se o certificado não estiver ativo,AWS IoTnão aceitará nenhuma tentativa de conexão que a use para autorização. Ao criar seu certificado, é fácil ignorar oAtivar. Felizmente, você pode ativar seu certificado daAWS IoTconsole.

Para verificar a ativação do certificado

1. NoAWS IoTconsoleNo menu esquerdo, escolhaSecuree, depois, escolhaCertificados.
2. Na lista de certificados, encontre o certificado que você criou para o exercício e verifique seu status naStatuscoluna.

   Se você não se lembrar do nome do certificado, verifique se há algum que sejaInativopara ver se eles podem ser o que você está usando.

   Selecione o certificado na lista para abrir sua página de detalhes. Na página de detalhes, você pode ver suaData de criaçãoPara ajudar a identificar o certificado.
3. Para ativar um certificado inativoNa página de detalhes do certificado, escolhaAçõese, depois, escolhaAtivar.

Se você encontrou o certificado correto e seu ativo, mas ainda estiver tendo problemas para executar o aplicativo de amostra, verifique sua política conforme a próxima etapa descreve.

Você também pode tentar criar uma coisa nova e um novo certificado seguindo as etapas emthe section called “Criar um objeto de coisa” (p. 41). Se você criar uma coisa nova, precisará dar a ela um nome novo e baixar os novos arquivos de certificado para o seu dispositivo.

Verifique a política anexada ao certificado

As políticas autorizam ações emAWS IoT. Se o certificado usado para se conectar aAWS IoTnão tem uma política, ou não tem uma política que permita que ela se conecte, a conexão será recusada, mesmo que o certificado esteja ativo.

Para verificar as políticas anexadas a um certificado

1. Encontre o certificado conforme descrito no item anterior e abra sua página de detalhes.
2. No menu à esquerda da página de detalhes do certificado, escolhaPolíticasPara ver as políticas anexadas ao certificado.
3. Se não houver políticas anexadas ao certificado, adicione uma escolhendo oAçõesmenu e, em seguida, escolherAnexar política.

   Escolha a política que você criou anteriormente nothe section called “CriarAWS IoTrecursos” (p. 39).
4. Se houver uma política anexada, escolha o bloco de política para abrir sua página de detalhes.

   Na página de detalhes, revise oDocumento de políticapara garantir que ele contenha as mesmas informações que a que você criouthe section called “Criar uma política do AWS IoT” (p. 40).

Verifique a linha de comando

Certifique-se de ter usado a linha de comando correta para o sistema. Os comandos usados em sistemas Linux/macOS geralmente são diferentes daqueles usados em sistemas Windows.

Verifique o endereço do endpoint

Revise o comando que você inseriu e verifique novamente o endereço do endpoint no comando para aquele em seuAWS IoTconsole.

Verifique os nomes dos arquivos dos arquivos de certificado

Compare os nomes dos arquivos no comando que você inseriu com os nomes de arquivo dos arquivos de certificado na`certs`directory.

Alguns sistemas podem exigir que os nomes dos arquivos estejam entre aspas para funcionar corretamente.

Verifique a instalação do SDK

Verifique se a instalação do SDK está completa e correta.

Em caso de dúvida, reinstale o SDK no seu dispositivo. Na maioria dos casos, isso é uma questão de encontrar a seção do tutorial intituladaInstalar aAWS IoTSDK de dispositivo da para***`Linguagem do SDK`***e seguindo o procedimento novamente.

Se você estiver usando oAWS IoTSDK de dispositivo da para JavaScript, lembre-se de instalar os aplicativos de amostra antes de tentar executá-los. A instalação do SDK não instala automaticamente os aplicativos de amostra. Os aplicativos de amostra devem ser instalados manualmente após a instalação do SDK.

Visualizar mensagens MQTT com o cliente MQTT do AWS IoT

Esta seção descreve como usar oAWS IoTCliente MQTT noAWS IoTconsolePara observar as mensagens MQTT enviadas e recebidas peloAWS IoT. O exemplo usado nesta seção está relacionado aos

exemplos usados emConceitos básicos do AWS IoT Core (p. 18)No entanto, você pode substituir o`topicName`usado nos exemplos com qualquernome do tópico ou filtro de tópico (p. 99)usado por sua solução de IoT.

Os dispositivos publicam mensagens MQTT que são identificadas porassuntos (p. 99)para comunicar seu estado paraAWS IoT, eAWS IoTO publica mensagens MQTT para informar os dispositivos e aplicativos sobre alterações e eventos. É possível usar o cliente MQTT para se inscrever nesses tópicos e observar as mensagens à medida que ocorrem. Você também pode usar o cliente MQTT para publicar mensagens MQTT em dispositivos e serviços inscritos noConta da AWS.

Visualizar mensagens MQTT no cliente MQTT

Como visualizar mensagens MQTT no cliente MQTT

1. NoAWS IoTconsoleNo menu esquerdo, escolhaTestee, depois, escolhaCliente de teste MQTT.

2. NoAssinar um tópicoguia, insira o`topicName`Para se inscrever no tópico no qual seu dispositivo publica. Para o aplicativo de exemplo de conceitos básicos, assine**#**, que se inscreve em todos os tópicos de mensagens.

   Continuando com o exemplo de introdução, noAssinar um tópicoguia, na guiaFiltro de tópicoscampo, insira**#**e, depois, escolhaSubscribe.

   A página de registro de mensagens do tópico,#Abre e**#**aparece noAssinaturasLista. Se o dispositivo que você configurou nothe section called “Configurar o dispositivo” (p. 43)está executando o programa de exemplo, você deve ver as mensagens que ele enviaAWS IoTno#log de mensagens. As entradas do registro de mensagens aparecerão abaixo doPublicarseção quando mensagens com o tópico inscrito são recebidas porAWS IoT.

3. No#página de registro de mensagens, você também pode publicar mensagens em um tópico, mas você precisará especificar o nome do tópico. Você não pode publicar no#tópico.

   As mensagens publicadas em tópicos inscritos aparecem no registro de mensagens à medida que são recebidas, com a mensagem mais recente primeiro.

Solução de problemas com mensagens MQTT

Use o filtro de tópicos curinga

Se suas mensagens não forem exibidas no registro de mensagens conforme esperado, tente se inscrever em um filtro de tópico curinga, conforme descrito emFiltros de tópicos (p. 100). O filtro de tópico curinga multi-nível MQTT é o sinal de hash ou libra (#) e pode ser usado como o filtro de tópicos naTópico de assinaturacampo.

Assinar o#filtro de tópicos se inscreve em todos os tópicos recebidos pelo corretor de mensagens. Você pode restringir o filtro substituindo elementos do caminho do filtro de tópico por um#caractere curinga de vários níveis ou o caractere wild-card de nível único '+'.

Ao usar curingas em um filtro de tópicos

- O caractere curinga de vários níveis deve ser o último caractere no filtro de tópico.
- O caminho do filtro de tópico pode ter apenas um caractere curinga de nível único por nível de tópico.

Por exemplo:

Filtro de tópicos	Exibe mensagens com
#	Nome de tópico
`topic_1/#`	Um nome de tópico que começa com`topic_1/`
`topic_1/level_2/#`	Um nome de tópico que começa com`topic_1/level_2/`
`topic_1/+/level_3`	Um nome de tópico que começa com`topic_1/`, termina com`/level_3`, e tem um elemento de qualquer valor no meio.

Para obter mais informações sobre os filtros de tópicos, consulteFiltros de tópicos (p. 100).

Verifique se há erros de nome de tópico

Os nomes de tópicos MQTT e os filtros de tópico diferenciam letras maiúsculas Se, por exemplo, o dispositivo estiver publicando mensagens para`Topic_1`(com capitalT) em vez de`topic_1`, tópico no qual você se inscreveu, suas mensagens não serão exibidas no cliente MQTT. Assinar o filtro de tópicos

curinga, no entanto, mostraria que o dispositivo está publicando mensagens e você pode ver que ele estava usando um nome de tópico que não era o que você esperava.

Publicar mensagens MQTT do cliente MQTT

Como publicar uma mensagem em um tópico MQTT

1. Na página do MQTT client (Cliente MQTT), naPublicar em um tópicoguia, na guiaNome do tópico, insira o*topicName*da sua mensagem. Neste exemplo, use **my/topic**.

   Note

   Não use informações de identificação pessoal em nomes de tópicos, seja usando-as no cliente MQTT ou na implementação do sistema. Os nomes dos tópicos podem aparecer em comunicações e relatórios não criptografados.

2. Na janela de carga da mensagem, insira o JSON a seguir:

```
{
    "message": "Hello, world",
    "clientType": "MQTT client"
}
```

3. SelecionePublicarPara publicar sua mensagem noAWS IoT.

   Note

   Verifique se você está inscrito nomy/topictópico antes de publicar sua mensagem.

4. NoAssinaturaslist, escolhamy/topicpara ver a mensagem. É necessário ver que a mensagem apareça no cliente MQTT abaixo da janela de carga útil da mensagem de publicação.

Você pode publicar mensagens MQTT em outros tópicos alterando o`topicName`noNome do tópicocampo e escolhendo oPublicar.

Como conectar-se ao AWS IoT Core

AWS IoT Coresuporta conexões com dispositivos IoT, gateways sem fio, serviços e aplicativos. Os dispositivos se conectam aoAWS IoT CorePara que eles possam enviar e receber dados doAWS IoTserviços e outros dispositivos. Aplicativos e outros serviços também se conectam aAWS IoT Corepara controlar e gerenciar os dispositivos IoT e processar os dados da sua solução de IoT. Esta seção descreve como escolher a melhor maneira de se conectar e se comunicar comAWS IoT Corepara cada aspecto da sua solução de IoT.

Há várias maneiras de interagir comAWS IoT. Aplicativos e serviços podem usar oAWS IoT Core- endpoints do plano de controle do (p. 71)e os dispositivos podem se conectar aAWS IoT Coreusando oAWS IoTendpoints de dispositivo (p. 72)ouAWS IoT Corepara gateways e dispositivos LoRaWAN (p. 73).

AWS IoT Core- endpoints do plano de controle do

OAWS IoT Core- plano de controleendpoints fornecem acesso a funções que controlam e gerenciam seuAWS IoTSolução.

- Endpoints

  OAWS IoT Core- plano de controleeAWS IoT CorePlano de controle Device Advisorendpoints são específicos da região e estão listados emAWS IoT CoreEndpoints e cotas do. Os formatos dos endpoints são os seguintes.

Objetivo do endpoint	Formato do endpoint	Serve
AWS IoT Core- plano de controle	`iot.aws-region.amazonaws.com`	AWS IoTAPI de plano de controle

Objetivo do endpoint	Formato do endpoint	Serve
AWS IoT CoreDevice Advisor - plano de controle	`api.iotdeviceadvisor.aws-region.amazonaws.com`	AWS IoT CoreAPI de plano de controle de dispositivo Advisor

- SDKs e ferramentas

  OAWSSDKs dafornecer suporte específico ao idioma para oAWS IoT CoreAPIs e as APIs de outrosAWSServiços da . OAWSSDKs móveis dofornecer aos desenvolvedores de aplicativos suporte específico da plataforma para oAWS IoT CoreAPI e outrosAWSServiços em dispositivos móveis.

  OAWS CLIfornece acesso à linha de comando às funções fornecidas peloAWS IoTendpoints de serviço do.AWS Tools for PowerShellfornece ferramentas para gerenciarAWSServiços e recursos no PowerShell Ambiente de script.

- Autenticação

  Os endpoints de serviço usam usuários do IAM eAWScredenciais para autenticar usuários.

- Saiba mais

  Para obter mais informações e links para referências do SDK, consultethe section called “Conexão aoAWS IoT CoreEndpoints de serviço da” (p. 73).

AWS IoTendpoints de dispositivo

OAWS IoTendpoints de dispositivos suportam comunicação entre seus dispositivos IoT eAWS IoT.

- Endpoints

  O suporte a endpoints do dispositivoAWS IoT CoreeAWS IoT Device ManagementFunções do . Eles são específicos doConta da AWSe você pode ver o que eles são usando odescribe-endpointcomando.

Objetivo do endpoint	Formato do endpoint	Serve
AWS IoT Core- plano de dados	Consulte ??? (p. 80).	AWS IoTAPI de plano de dados
AWS IoT Device Management- dados de trabalhos	Consulte ??? (p. 80).	AWS IoTPlano de dados de tarefas API
AWS IoTDevice Advisor - plano de dados	Consulte ??? (p. 1070).	
AWS IoT Device Management- Fleet Hub		
AWS IoT Device Management- Encapsulamento seguro do	`api.tunneling.iot.aws-region.amazonaws.com`	AWS IoTAPI de encapsulamento seguro

  Para obter mais informações sobre esses endpoints e as funções que eles oferecem suporte, consultethe section called “AWS IoTdados do dispositivo e endpoints de serviço” (p. 80).

- SDKs

  OAWS IoTSDKs de dispositivo da (p. 81)fornecer suporte específico ao idioma para o Transporte de Telemetria de Enfileiramento de Mensagens (MQTT) e WebSocket Protocolos seguros (WSS), com os quais dispositivos usam para se comunicarAWS IoT.SDKs móveis do AWS (p. 78)também fornecer

suporte para comunicações de dispositivos MQTT,AWS IoTAPIs e as APIs de outrosAWSServiços em dispositivos móveis.

- Autenticação

  Os endpoints do dispositivo usam certificados X.509 ouAWSUsuários do IAM com credenciais para autenticar usuários.

- Saiba mais

  Para obter mais informações e links para referências do SDK, consultethe section called “SDKs de dispositivo da AWS IoT” (p. 81).

AWS IoT Corepara gateways e dispositivos LoRaWAN

AWS IoT Corepara LoRaWAN conecta gateways e dispositivos sem fio aAWS IoT Core.

- Endpoints

  AWS IoT Corefor LoraWAN gerencia as conexões de gateway para conta e região específicaAWS IoT CoreEndpoints do . Os gateways podem se conectar ao endpoint CUPS (Configuration and Update Server) da sua conta queAWS IoT Corepara LoRaWAN fornece.

Objetivo do endpoint	Formato do endpoint	Serve
Servidor de configuração e atualização (CUPS)	`account-specific-prefix.cups.lorawan.aws-region.amazonaws.com:443`	Comunicação de gateway com o Servidor de Configuração e Atualização fornecido porAWS IoT Corepara LoRaWAN
Servidor de rede LoraWAN (LNS)	`account-specific-prefix.gateway.lorawan.aws-region.amazonaws.com:443`	Comunicação de gateway com o LoraWAN Network Server fornecido porAWS IoT Corepara LoRaWAN

- SDKs

  OAWS IoTAPI sem fio queAWS IoT Corepara LorAwan é construído em é suportado peloAWSSDK. Para obter mais informações, consulteAWSSDKs e Toolkits.

- Autenticação

  AWS IoT CorePara comunicações de dispositivo LoRawan use certificados X.509 para proteger comunicações com oAWS IoT.

- Saiba mais

  Para obter mais informações sobre como configurar e conectar dispositivos sem fio, consulteAWS IoT Corepara LoRaWAN (p. 1126).

Conexão aoAWS IoT CoreEndpoints de serviço da

Acesse os recursos doAWS IoT Core- plano de controleusando oAWS CLI, oAWSSDK para seu idioma preferido ou chamando a API REST diretamente. Recomendamos usar oAWS CLIou umAWSSDK para interagir comAWS IoT CorePorque eles incorporam as melhores práticas para ligarAWSServiços da .

Chamar as APIs REST diretamente é uma opção, mas você deve fornecerAs credenciais de segurança necessáriasque permitem acesso à API.

> Note
>
> Dispositivos de IoT devem usarSDKs de dispositivo da AWS IoT (p. 81). Os SDKs do dispositivo são otimizados para uso em dispositivos, suportam comunicação MQTT comAWS IoT, e suporte aoAWS IoTAPIs mais usadas por dispositivos. Para obter mais informações sobre os SDKs de dispositivos e os recursos que eles fornecem, consulteSDKs de dispositivo da AWS IoT (p. 81).
> Dispositivos móveis devem usarSDKs móveis do AWS (p. 78). Os SDKs móveis oferecem suporte paraAWS IoTAPIs, comunicações de dispositivos MQTT e as APIs de outrosAWSServiços em dispositivos móveis. Para obter mais informações sobre os SDKs móveis e os recursos que eles fornecem, consulteSDKs móveis do AWS (p. 78).

Você pode usarAWS Amplifyferramentas e recursos em aplicativos web e móveis para se conectar mais facilmenteAWS IoT Core. Para obter mais informações sobre como se conectar aoAWS IoT Coreusando o Amplify, consulteConceitos básicos do Pub Subna documentação do Amplify.

As seções a seguir descrevem as ferramentas e SDKs que você pode usar para desenvolver e interagir com oAWS IoTe outrosAWSServiços da . Para obter a lista completa deAWSferramentas e kits de desenvolvimento disponíveis para criar e gerenciar aplicativos emAWS, consulteFerramentas para construirAWS.

AWS CLI para AWS IoT Core

OAWS CLIfornece acesso à linha de comando aoAWSAPIs.

- Instalação

  Para obter informações sobre como instalar oAWS CLI, consulteInstalar oAWS CLI.
- Autenticação

  OAWS CLIUsar credenciais de seuConta da AWS.
- Referência

  Para obter informações sobre oAWS CLIcomandos para estesAWS IoT Coreserviços, consulte:
  - AWS CLIReferência de comando para IoT
  - AWS CLIReferência de comando para dados de IoT
  - AWS CLIReferência de comando para dados de trabalhos de IoT
  - AWS CLIReferência de comando para tunelamento seguro de IoT

Para ferramentas para gerenciarAWSServiços e recursos no PowerShell ambiente de script, consulteAWSTools for PowerShell.

AWS SDKs

comAWSSDKs, seus aplicativos e dispositivos compatíveis podem chamarAWS IoTAPIs e as APIs de outrosAWSServiços da . Esta seção fornece links para aAWSSDKs e a documentação de referência da API das APIs doAWS IoT CoreServiços da .

OAWSSDKs suportam essesAWS IoT CoreAPIs

- AWS IoT
- AWS IoTPlano de dados do

- AWS IoTPlano de dados do trabalho
- AWS IoTEncapsulamento seguro do
- AWS IoTSem fio

C++

Para instalar oAWS SDK for C++e use-o para conectar-se aoAWS IoT:

1. Siga as instruções emConceitos básicos do uso daAWSSDK for C++

   Estas instruções descrevem como:
   - Instalar e compilar o SDK a partir de arquivos de origem
   - Forneça credenciais para usar o SDK com seuConta da AWS
   - Inicialize e desligue o SDK em seu aplicativo ou serviço
   - Crie um projeto CMake para criar seu aplicativo ou serviço
2. Crie e execute um aplicativo de amostra. Para aplicativos de amostra que usam oAWSSDK for C++, consulteAWS SDK for C++Exemplos de código.

Documentação para oAWS IoT CoreServiços que oAWS SDK for C++aguenta

- Documentação de referência do Aws# IoT# IoT Client
- Documentação de referência Aws# IOTDataPlane# IOTDataPlaneClient
- Aws# IOTJobsDataPlane# Documentação de referência IOTJobsDataPlaneClient
- Aws# IOTSecureTunneling# Documentação de referência IOTSecureTunnelingClient

Go

Para instalar oAWS SDK for Goe use-o para conectar-se aoAWS IoT:

1. Siga as instruções emConceitos básicos doAWS SDK for Go

   Estas instruções descrevem como:
   - Instalar a AWS SDK for Go
   - Obtenha chaves de acesso para o SDK para acessar seuConta da AWS
   - Importe pacotes para o código-fonte de nossos aplicativos ou serviços
2. Crie e execute um aplicativo de amostra. Para aplicativos de amostra que usam oAWS SDK for Go, consulteAWS SDK for GoExemplos de código.

Documentação para oAWS IoT CoreServiços que oAWS SDK for Goaguenta

- Documentação de referência da IoT
- Documentação de referência IOTDataPlane
- Documentação de referência IOTJobsDataPlane
- Documentação de referência do iOTSecureUnneling

Java

Para instalar oAWS SDK for Javae use-o para conectar-se aoAWS IoT:

1. Siga as instruções emConceitos básicos doAWS SDK for Java 2.x

Estas instruções descrevem como:

- Cadastro naAWSe Criar um usuário do IAM
- Fazer download do SDK
- ConfigurarAWSCredenciais e região
- Usar o SDK com o Apache Maven
- Usar o SDK com o Gradle

2. Crie e execute um aplicativo de amostra usando um dosAWS SDK for Java 2.xExemplos de código.
3. Analise oDocumentação de referência da API do SDK

Documentação para oAWS IoT CoreServiços que oAWS SDK for Javaaguenta

- Documentação de referência do IOTClient
- Documentação de referência iOTDataPlaneClient
- Documentação de referência IOTJobsDataPlaneClient
- Documentação de referência iOTSecureTunnelingClient

JavaScript

Para instalar oAWS SDK for JavaScripte use-o para conectar-se aoAWS IoT:

1. Siga as instruções emConfigurar oAWS SDK for JavaScript. Essas instruções se aplicam ao usar aAWS SDK for JavaScriptno navegador e com Node.JS. Certifique-se de seguir as instruções que se aplicam à sua instalação.

   Estas instruções descrevem como:

   - Verifique os pré-requisitos
   - Instalar o SDK para JavaScript
   - Carregar o SDK para JavaScript

2. Crie e execute um aplicativo de amostra para começar a usar o SDK, conforme a opção de introdução para seu ambiente descreve.
   - Conceitos básicos doAWSSDK para o JavaScript No Navegador, ou
   - Conceitos básicos doAWSSDK para o JavaScript no Node.js

Documentação para oAWS IoT CoreServiços que oAWS SDK for JavaScriptaguenta

- `AWS.Iot reference documentation`
- `AWS.IotData reference documentation`
- `AWS.IotJobsDataPlane reference documentation`
- `AWS.IotSecureTunneling reference documentation`

.NET

Para instalar oAWS SDK for .NETe use-o para conectar-se aoAWS IoT:

1. Siga as instruções emConfiguração doAWS SDK for .NETmeio Ambiente
2. Siga as instruções emConfiguração doAWS SDK for .NETprojeto

   Estas instruções descrevem como:

- Inicie um novo projeto
- Obter e configurarAWScredenciais
- InstallAWSPacotes SDK

3. Crie e execute um dos programas de amostra noTrabalho comAWSServiços noAWSSDK for .NET
4. Analise oDocumentação de referência da API do SDK

Documentação para oAWS IoT CoreServiços que oAWS SDK for .NETaguenta

- Documentação de referência da Amazon.iot.model
- Documentação de referência da Amazon.iotData.model
- Documentação de referência Amazon.iotJobsDataPlane.model
- Documentação de referência da Amazon.iotSecureTunneling.model

PHP

Para instalar oAWS SDK for PHPe use-o para conectar-se aoAWS IoT:

1. Siga as instruções emConceitos básicos doAWS SDK for PHPVersão 3

   Estas instruções descrevem como:
   - Verifique os pré-requisitos
   - Instalar o SDK
   - Aplicar o SDK a um script PHP
2. Crie e execute um aplicativo de amostra usando um dosAWS SDK for PHPExemplos de código da versão 3

Documentação para oAWS IoT CoreServiços que oAWS SDK for PHPaguenta

- Documentação de referência do IOTClient
- Documentação de referência iOTDataPlaneClient
- Documentação de referência IOTJobsDataPlaneClient
- Documentação de referência iOTSecureTunnelingClient

Python

Para instalar oAWS SDK for Python (Boto3)e use-o para conectar-se aoAWS IoT:

1. Siga as instruções noAWS SDK for Python (Boto3)Início rápido

   Estas instruções descrevem como:
   - Instalar o SDK
   - Configurar o SDK
   - Usar o SDK em seu código
2. Criar e executar um programa de exemplo que use aAWS SDK for Python (Boto3)

   Este programa exibe as opções de registro atualmente configuradas da conta. Depois de instalar o SDK e configurá-lo para sua conta, você deve conseguir executar este programa.

```
import boto3
import json
```

```
# initialize client
iot = boto3.client('iot')

# get current logging levels, format them as JSON, and write them to stdout
response = iot.get_v2_logging_options()
print(json.dumps(response, indent=4))
```

Para obter mais informações sobre a função usada neste exemplo, consultethe section called “Configurar registro em log da AWS IoT” (p. 433).

Documentação para oAWS IoT CoreServiços que oAWS SDK for Python (Boto3)aguenta

- Documentação de referência da IoT
- Documentação de referência IOTDataPlane
- Documentação de referência IOTJobsDataPlane
- Documentação de referência do iOTSecureUnneling

Ruby

Para instalar oAWS SDK for Rubye use-o para conectar-se aoAWS IoT:

- Siga as instruções emConceitos básicos doAWS SDK for Ruby

  Estas instruções descrevem como:
  - Instalar o SDK
  - Configurar o SDK
- Criar e executar oTutorial Hello World

Documentação para oAWS IoT CoreServiços que oAWSSuportes para SDK for Ruby

- Documentação de referência do Aws# IoT# Client
- Aws# IOTDataPlane# Documentação de referência do cliente
- Aws# IOTJobsDataPlane# Documentação de referência do cliente
- Aws# IOTSecureTunneling# Documentação de referência do cliente

SDKs móveis do AWS

OAWSOs SDKs móveis oferecem suporte específico da plataforma para desenvolvedores de aplicativos móveis para as APIs doAWS IoT Coreserviços, comunicação de dispositivos IoT usando MQTT e as APIs de outrosAWSServiços da .

Android

AWS Mobile SDK for Android

OAWS Mobile SDK for Androidcontém uma biblioteca, exemplos e documentação para que os desenvolvedores criem aplicativos móveis conectados usandoAWS. Esse SDK também inclui suporte para comunicações de dispositivo MQTT e chamando as APIs daAWS IoT CoreServiços da . Para obter mais informações, consulte as informações a seguir:

- AWSMobile SDK for Android no GitHub

- AWSLeia-me do SDK móvel para Android
- AWSExemplos do SDK móvel para Android
- AWSReferência à API SDK for Android
- Documentação de referência da AwsiotClient Class

iOS

AWS Mobile SDK for iOS

OAWS Mobile SDK for iOSÉ um kit de desenvolvimento de software de código aberto distribuído sob uma licença Apache Open Source. O SDK for iOS fornece uma biblioteca, exemplos de código e documentação para ajudar os desenvolvedores a criar aplicativos móveis conectados usandoAWS. Esse SDK também inclui suporte para comunicações de dispositivo MQTT e chamando as APIs daAWS IoT CoreServiços da . Para obter mais informações, consulte as informações a seguir:

- AWS Mobile SDK for iOS no GitHub
- AWSReadme do SDK for iOS
- AWSExemplos do SDK for iOS
- AWS IoTDocumentos de referência de classe noAWSSDK do para iOS

APIs REST doAWS IoT Coreserviços

As APIs REST doAWS IoT Coreserviços podem ser chamados diretamente usando solicitações HTTP.

- URL do endpoint

  Os endpoints de serviço que expõem as APIs REST doAWS IoT Coreos serviços variam de acordo com a Região e estão listados emAWS IoT CoreEndpoints e cotas do. Você deve usar o endpoint para a região que tem oAWS IoTRecursos que você deseja acessar, porqueAWS IoTOs recursos do são específicos da região.
- Autenticação

  As APIs REST doAWS IoT CoreUtilização de serviçosAWSCredenciais do IAM para autenticação. Para obter mais informações, consulteAssinaturaAWSSolicitações da API donoAWSReferência geral do.
- Referência de API

  Para obter informações sobre as funções específicas fornecidas pelas APIs REST doAWS IoT Coreserviços, consulte:
  - Referência da API para IoT.
  - Referência de API para dados de IoT.
  - Referência de API para dados de trabalhos de IoT.
  - Referência de API para tunelamento seguro de IoT.

Conectar dispositivos aoAWS IoT

Os dispositivos se conectam aoAWS IoTe outros serviços por meio deAWS IoT Core. Por meio doAWS IoT Core, os dispositivos enviam e recebem mensagens usando endpoints de dispositivos específicos da sua conta. Othe section called “SDKs de dispositivo da AWS IoT” (p. 81)suporte a comunicações de dispositivos usando os protocolos MQTT e WSS. Para obter mais informações sobre os protocolos que os dispositivos podem usar, consultethe section called “Protocolos de comunicação de” (p. 83).

O agente de mensagens

AWS IoTgerencia a comunicação do dispositivo por meio de um corretor de mensagens. Dispositivos e clientes publicam mensagens no corretor de mensagens e também assinam mensagens publicadas pelo corretor de mensagens. As mensagens são identificadas por um aplicativo definidotópico (p. 99). Quando o agente de mensagens recebe uma mensagem publicada por um dispositivo ou cliente, ele republica essa mensagem nos dispositivos e clientes que assinaram o tópico da mensagem. O corretor de mensagens também encaminha mensagens para oAWS IoT regras (p. 478)mecanismo, que pode atuar no conteúdo da mensagem.

AWS IoTSegurança de mensagens

Conexões de dispositivo paraAWS IoTusathe section called “Certificados do cliente X.509” (p. 298)eAWSAssinatura V4Para autenticação. As comunicações do dispositivo são protegidas pelo TLS versão 1.2 eAWS IoTrequer dispositivos para enviar oExtensão da Server Name Indication (SNI)quando eles se conectam. Para obter mais informações, consulte Segurança de transporte no AWS IoT.

AWS IoTdados do dispositivo e endpoints de serviço

Cada conta tem vários endpoints de dispositivo exclusivos da conta e suportam funções de IoT específicas. OAWS IoTos endpoints de dados do dispositivo suportam um protocolo de publicação/assinatura projetado para as necessidades de comunicação dos dispositivos IoT; no entanto, outros clientes, como aplicativos e serviços, também podem usar essa interface se seu aplicativo exigir os recursos especializados que esses endpoints fornecem. OAWS IoTendpoints de serviço de dispositivo oferecem suporte ao acesso centrado no dispositivo a serviços de segurança e gerenciamento.

Para aprender o endpoint de dados do dispositivo da sua conta, você pode encontrá-lo noConfigurações Página do seuAWS IoT Coreconsole do .

Para aprender o endpoint do dispositivo da sua conta para uma finalidade específica, incluindo o endpoint de dados do dispositivo, use odescribe-endpointComando da CLI mostrado aqui, ou o`DescribeEndpoint`API REST e forneça o*`endpointType`*Valor de parâmetro da tabela a seguir.

```
aws iot describe-endpoint --endpoint-type endpointType
```

Esse comando retorna um*`iot-endpoint`*no seguinte formato:*`account-specific-prefix`*`.iot.`*`aws-region`*`.amazonaws.com`.

O`DescribeEndpoint`A API não precisa ser consultada toda vez que um novo dispositivo estiver conectado. Os endpoints que você cria persistem para sempre e não mudam depois de criados.

Todo cliente tem um`iot:Data-ATS`e um`iot:Data`. Cada endpoint usa um certificado X.509 para autenticar o cliente. É altamente recomendável que os clientes usem o tipo de endpoint `iot:Data-ATS` mais recente a fim de evitar problemas relacionados ao próximo cancelamento de confiança difundido pelas autoridades de certificação do Symantec. Nós fornecemos o`iot:Data`endpoint para dispositivos recuperarem dados de endpoints antigos que usam VeriSign Certificados para compatibilidade com versões anteriores. Para obter mais informações, consulte Autenticação do servidor.

AWS IoTendpoints para dispositivos

Objetivo do endpoint	***`endpointType`*** value	Descrição
AWS IoT Core- Operações de plano de dados	`iot:Data-ATS`	Usado para enviar e receber dados de e para o agente de mensagens,Device Shadow (p. 633),

Objetivo do endpoint	*endpointType* value	Descrição
		eMecanismo de regras do (p. 478)componentes deAWS IoT. `iot:Data-ATS`retorna um endpoint de dados assinados pela ATS.
AWS IoT Core- operações de plano de dados (legado)	`iot:Data`	`iot:Data`retorna um VeriSign endpoint de dados assinado fornecido para compatibilidade com versões anteriores.
AWS IoT CoreAcesso credencial	`iot:CredentialProvider`	Usado para trocar um certificado X.509 integrado de um dispositivo para credenciais temporárias para conectar-se diretamente a outrosAWSServiços da . Para obter mais informações sobre como conectar-se a outros serviços da AWS, consulte Autorizar chamadas diretas para serviços da AWS (p. 375).
AWS IoT Device Management- operações de dados de trabalhos	`iot:Jobs`	Usado para permitir que dispositivos interajam com aAWS IoTServiço Jobs usando oAPIs HTTPS do dispositivo de trabalho (p. 756).
AWS IoTOperações Device Advisor	`iot:DeviceAdvisor`	Um tipo de endpoint de teste usado para testar dispositivos com o Device Advisor. Para obter mais informações, consulte ??? (p. 1067).
AWS IoT Coredados beta (pré-visualização)	`iot:Data-Beta`	Um tipo de endpoint reservado para versões beta. Para obter informações sobre o uso atual, consulte??? (p. 117).

Você também pode usar seu próprio nome de domínio totalmente qualificado (FQDN), como*`example.com`*e o certificado de servidor associado ao qual conectar dispositivosAWS IoTUsando othe section called “endpoints configuráveis” (p. 117).

SDKs de dispositivo da AWS IoT

OAWS IoTOs SDKs de dispositivos ajudam você a conectar seus dispositivos IoT aAWS IoT Coree eles suportam protocolos MQTT e MQTT sobre WSS.

OAWS IoTOs SDKs do dispositivo diferem dosAWSSDKs em que oAWS IoTOs SDKs de dispositivos suportam as necessidades de comunicação especializadas dos dispositivos IoT, mas não suportam todos os serviços suportados peloAWSSDKs. OAWS IoTOs SDKs do dispositivo são compatíveis com oAWSSDKs que suportam todos osAWSserviços; no entanto, eles usam métodos de autenticação

diferentes e se conectam a diferentes endpoints, o que poderia fazer o uso doAWSSDKs impraticáveis em um dispositivo IoT.

Dispositivos móveis

Othe section called “SDKs móveis do AWS” (p. 78)suporta ambas as comunicações do dispositivo MQTT, algumas dasAWS IoTAPIs de serviço e as APIs de outrosAWSServiços da . Se você estiver desenvolvendo em um dispositivo móvel compatível, revise o SDK para ver se ele é a melhor opção para desenvolver sua solução de IoT.

C++

AWS IoTSDK de dispositivo C++ da

OAWS IoTO SDK de dispositivo C++ da permite que os desenvolvedores compilem aplicativos conectados usandoAWSe as APIs doAWS IoT CoreServiços da . Esse SDK foi especificamente projetado para dispositivos que não têm restrições de recursos e exigem recursos avançados, como enfileiramento de mensagens, suporte a vários threads e os mais recentes recursos de linguagem. Para mais informações, consulte:

- AWS IoTSDK de dispositivo C++ v2 no GitHub
- AWS IoTLeia-me do SDK do dispositivo C++ v2
- AWS IoTExemplos de SDK de dispositivo C++ v2
- AWS IoTDocumentação da API do SDK C++ v2 do dispositivo

Python

AWS IoTSDK de dispositivo da para Python

OAWS IoTO SDK de dispositivo da para Python permite que os desenvolvedores escrevam scripts Python para usar seus dispositivos para acessar oAWS IoTplataforma através de MQTT ou MQTT sobre o WebSocket Protocolo seguro (WSS). Conectando seus dispositivos às APIs doAWS IoT CoreOs usuários podem trabalhar de modo seguro com o agente de mensagens, as regras e o serviço Device Shadow queAWS IoT Corefornece e com outrosAWSServiços comoAWS Lambda, Amazon Kinesis e Amazon S3 e muito mais.

- SDK de dispositivo AWS IoT para Python v2 no GitHub
- AWS IoT Device SDK for Python v2 Readme
- AWS IoTSDK de dispositivo para amostras do Python v2
- AWS IoTDocumentação da API do SDK de dispositivo para Python v2

JavaScript

AWS IoTSDK de dispositivo da para JavaScript

OAWS IoTSDK de dispositivo da para JavaScript torna possível que os desenvolvedores escrevam JavaScript aplicativos que acessam APIs doAWS IoT CoreUsar MQTT ou MQTT pelo WebSocket Protocol. Ele pode ser usado em ambientes Node.js e aplicações de navegador. Para obter mais informações, consulte as informações a seguir:

- AWS IoTSDK de dispositivo da para JavaScript V2 no GitHub
- AWS IoTSDK de dispositivo da para JavaScript Leiame do v2
- AWS IoTSDK de dispositivo da para JavaScript Amostras do v2
- AWS IoTSDK de dispositivo da para JavaScript Documentação da API v2

Java

AWS IoTSDK de dispositivo da para Java

OAWS IoTO SDK de dispositivo da para Java permite que os desenvolvedores de Java acessem as APIs daAWS IoT CorePor meio de MQTT ou MQTT pelo protocolo WebSocket. O SDK oferece suporte ao serviço Device Shadow. Você pode acessar as sombras usando métodos HTTP, inclusive GET, UPDATE e DELETE. O SDK também oferece suporte a um modelo de acesso simplificado de shadow, o que permite que os desenvolvedores troquem dados com shadows usando os métodos getter e setter, sem necessidade de serializar ou desserializar nenhum documento JSON. Para obter mais informações, consulte as informações a seguir:

- SDK de dispositivo AWS IoT para Java v2 no GitHub
- Leia-me do SDK de dispositivo AWS IoT para Java v2
- AWS IoTExemplos de SDK de dispositivo para Java v2
- AWS IoTDocumentação da API do SDK de dispositivo para Java v2

Embedded C

AWS IoTSDK de dispositivo da para C incorporado

> **Important**
>
> Este SDK destina-se ao uso por desenvolvedores experientes de software incorporado.

OAWS IoT Device SDK para C incorporado(C-SDK) é um conjunto de arquivos de origem C sob a licença de código aberto do MIT que pode ser usado em aplicativos incorporados para conectar dispositivos de IoT de modo seguro aAWSIoT Core. Inclui MQTT, JSON Parser eAWS IoTBibliotecas Device Shadow e outras. Ele é distribuído na forma de origem e deve ser incorporado ao firmware do cliente junto com o código do aplicativo, outras bibliotecas e, opcionalmente, um RTOS (Sistema Operacional em Tempo Real).

Em geral, o AWS IoT Device SDK para C incorporado destina-se a dispositivos com restrição de recursos que exigem um tempo de execução de linguagem C otimizado. É possível usar o SDK em qualquer sistema operacional e hospedá-lo em qualquer tipo de processador (p. ex., MCUs e MPUs). Se o dispositivo tiver recursos de processamento e memória suficientes disponíveis, recomendamos que você use um dos outrosAWS IoTSDKs de dispositivo e dispositivos móveis, como oAWS IoTSDK de dispositivo para C++, Java, JavaScript ou Python.

Para obter mais informações, consulte as informações a seguir:

- SDK de dispositivo AWS IoT para C incorporado no GitHub
- Arquivo leiame do SDK de dispositivo da AWS IoT para C incorporado
- AWS IoTSDK de dispositivo para amostras C incorporadas

Protocolos de comunicação de

AWS IoT Coresuporta dispositivos e clientes que usam o MQTT e o MQTT sobre WebSocket Protocolos seguros (WSS) para publicar e assinar mensagens e dispositivos e clientes que usam o protocolo HTTPS para publicar mensagens. Todos os protocolos suportam IPv4 e IPv6. Esta seção descreve as diferentes opções de conexão para dispositivos e clientes.

TLS v1.2

AWS IoT CoreusaTLS versão 1.2para criptografar toda a comunicação. Os clientes também devem enviar oExtensão TLS da Server Name Indication (SNI). Tentativas de conexão que não incluem o SNI são recusadas. Para obter mais informações, consulte Segurança de transporte no AWS IoT.

OSDKs de dispositivo da AWS IoT (p. 81)suporte MQTT e MQTT sobre WSS e suporta os requisitos de segurança das conexões do cliente. Recomendamos usar oSDKs de dispositivo da AWS IoT (p. 81)para conectar clientes aAWS IoT.

Protocolos, mapeamentos de porta e autenticação

A forma como um dispositivo ou cliente se conecta ao agente de mensagens usando um endpoint de dispositivo depende do protocolo que ele usa. A tabela a seguir lista os protocolos que oAWS IoTsuporte a endpoints de dispositivo e os métodos de autenticação e portas que eles usam.

Protocolos, autenticação e mapeamentos de porta

Protocolo	Operações compatíveis	Autenticação	Port	Nome do protocolo ALPN
MQTT pelo WebSocket	Publicar, assinar	Signature versão 4	443	N/D
MQTT pelo WebSocket	Publicar, assinar	Autenticação personalizada	443	N/D
MQTT	Publicar, assinar	Certificado do cliente X.509	443†	`x-amzn-mqtt-ca`
MQTT	Publicar, assinar	Certificado do cliente X.509	8883	N/D
MQTT	Publicar, assinar	Autenticação personalizada	443†	`mqtt`
HTTPS	Publicar somente	Signature versão 4	443	N/D
HTTPS	Publicar somente	Certificado do cliente X.509	443†	`x-amzn-http-ca`
HTTPS	Publicar somente	Certificado do cliente X.509	8443	N/D
HTTPS	Publicar somente	Autenticação personalizada	443	N/D

Negociação de protocolo de camada de aplicativos (ALPN)

†Os clientes que se conectam na porta 443 com autenticação de certificado de cliente X.509 devem implementar oNegociação de protocolo de camada de aplicativos (ALPN)Extensão TLS e use oNome do protocolo ALPNlistado no ALPN ProtocolNameList enviado pelo cliente como parte do`ClientHello`mensagem.
Na porta 443, oIOT: Data-ATS (p. 80)O endpoint oferece suporte a ALPN x-amzn-http-ca HTTP, mas oIOT:Jobs (p. 80)endpoint não.
Na porta 8443 HTTPS e porta 443 MQTT com ALPN x-amzn-mqtt-ca,Autenticação personalizada (p. 319)As não podem ser usadas.

Os clientes se conectam aos seusConta da AWSendpoints do dispositivo. Consultethe section called “AWS IoTdados do dispositivo e endpoints de serviço” (p. 80)Para obter informações sobre como encontrar os endpoints do dispositivo da sua conta.

Note

AWSOs SDKs não exigem o URL inteiro. Eles só exigem o nome do host do endpoint, como o`pubsub.py`Exemplo do para oAWS IoTSDK de dispositivo da para Python no GitHub. Passar

o URL inteiro conforme fornecido na tabela a seguir pode gerar um erro, como nome de host inválido.

Como conectar-se ao AWS IoT Core

Protocolo	Endpoint ou URL
MQTT	`iot-endpoint`
MQTT pelo WSS	`wss://iot-endpoint/mqtt`
HTTPS	`https://iot-endpoint/topics`

Escolhendo um protocolo para a comunicação do seu dispositivo

Para a maioria das comunicações de dispositivos IoT através dos endpoints do dispositivo, você vai querer usar os protocolos MQTT ou MQTT sobre WSS; no entanto, os endpoints do dispositivo também suportam HTTPS. A tabela a seguir compara comoAWS IoT Coreusa os dois protocolos para comunicação do dispositivo.

AWS IoTProtocolos de dispositivo lado a lado

Recurso	MQTT (p. 86)	HTTPS (p. 97)
Suporte para publicar/assinar	Publicar e assinar	Publicar somente
Compatibilidade com o SDK	AWSSDKs de dispositivo da (p. 81)suporte a protocolos MQTT e WSS	Não há suporte a SDK, mas você pode usar métodos específicos de idioma para fazer solicitações HTTPS
Suporte à qualidade de serviço	MQTT QoS níveis 0 e 1 (p. 88)	O QoS é suportado pela passagem de um parâmetro de string de consulta`?qos=qos`onde o valor pode ser 0 ou 1. Você pode adicionar essa string de consulta para publicar uma mensagem com o valor de QoS desejado.
Pode receber mensagens perdidas enquanto o dispositivo estava offline	Sim	Não
`clientId`suporte de campo	Sim	Não
Detecção de desconexão do dispositivo	Sim	Não
Comunicação segura	Sim. Consulte Protocolos, mapeamentos de porta e autenticação (p. 84)	Sim. Consulte Protocolos, mapeamentos de porta e autenticação (p. 84)
Definições de tópicos	Aplicativo definido	Aplicativo definido
Formato de dados de mensagens	Aplicativo definido	Aplicativo definido
Sobrecarga do protocolo	LOWER	Mais alto

Recurso	MQTT (p. 86)	HTTPS (p. 97)
Consumo de energia	LOWER	Mais alto

Limites de duração da conexão

As conexões HTTPS não são garantidas para durar mais do que o tempo necessário para receber e responder às solicitações.

A duração da conexão MQTT depende do recurso de autenticação que você usa. A tabela a seguir lista a duração máxima da conexão em condições ideais para cada recurso.

Duração da conexão MQTT por recurso de autenticação

Recurso	Duração máxima*
Certificado do cliente X.509	1 a 2 semanas
Autenticação personalizada	1 a 2 semanas
Signature versão 4	Até 24 horas

*Não garantido

Com certificados X.509 e autenticação personalizada, a duração da conexão não tem limite rígido, mas pode ser tão curta quanto alguns minutos. Interrupções de conexão podem ocorrer por vários motivos. A lista a seguir contém alguns dos motivos mais comuns.

- Interrupções de disponibilidade do Wi-Fi
- Interrupções de conexão do provedor de serviços de Internet (ISP)
- Patches de serviço
- Implantações de serviços
- Autoescalabilidade do serviço
- Host de serviço indisponível
- Problemas e atualizações do load balancer
- Erros no lado do cliente,

Seus dispositivos devem implementar estratégias para detectar desconexões e reconectar. Para obter informações sobre eventos de desconexão e orientações sobre como tratá-los, consulte??? (p. 1121)em??? (p. 1121).

MQTT

MQTTÉ um protocolo de sistema de mensagens leve amplamente adotado projetado para dispositivos com restrições.AWS IoTsuporte para MQTT é baseado noEspecificação MQTT v3.1.1, com algumas diferenças. Para obter informações sobre comoAWS IoTDifere da especificação MQTT v3.1.1, consultethe section called “AWS IoTDiferenças da especificação MQTT versão 3.1.1” (p. 96).

AWS IoT Coresuporta conexões de dispositivo que usam o protocolo MQTT e o protocolo MQTT sobre WSS e que são identificados por umID de cliente. OSDKs de dispositivo da AWS IoT (p. 81)suporta ambos os protocolos e são as formas recomendadas de conectar dispositivos aAWS IoT. OAWS IoTOs SDKs de dispositivos suportam as funções necessárias para que dispositivos e clientes se conectem e

acessemAWS IoT CoreServiços da . Os Device SDKs suportam os protocolos de autenticação que oAWS IoTos serviços exigem e os requisitos de ID de conexão que o protocolo MQTT e os protocolos MQTT sobre WSS exigem. Para obter informações sobre como se conectar aoAWS IoTUsando oAWSSDKs de dispositivos e links para exemplos deAWS IoTNos idiomas compatíveis, consultethe section called "Conectando com o MQTT usando oAWS IoTSDKs de dispositivo da" (p. 87). Para obter mais informações sobre os métodos de autenticação e os mapeamentos de porta para mensagens MQTT, consulteProtocolos, mapeamentos de porta e autenticação (p. 84).

Embora recomendamos usar oAWS IoTSDKs de dispositivo para se conectarAWS IoT, eles não são obrigatórios. Se você não usar oAWS IoTOs SDKs do dispositivo, no entanto, você deve fornecer a segurança de conexão e comunicação necessária. Os clientes devem enviar oExtensão TLS da Server Name Indication (SNI)na solicitação de conexão. Tentativas de conexão que não incluem o SNI são recusadas. Para obter mais informações, consulte Segurança de transporte no AWS IoT. Clientes que usam usuários do IAM eAWScredenciais para autenticar clientes devem fornecer o corretoSignature versão 4Autenticação.

Conectando com o MQTT usando oAWS IoTSDKs de dispositivo da

Esta seção contém links para aAWS IoTSDKs de dispositivos e para o código-fonte de programas de amostra que ilustram como conectar um dispositivo aAWS IoT. Os aplicativos de amostra vinculados aqui mostram como se conectar aoAWS IoTUsando o protocolo MQTT e MQTT sobre WSS.

C++

Usar oAWS IoTSDK de dispositivo C++ para conectar dispositivos

- Código-fonte de um aplicativo de exemplo que mostra um exemplo de conexão MQTT em C++
- SDK de dispositivo C++ AWS IoT v2 no GitHub

Python

Usar oAWS IoTSDK de dispositivo para Python para conectar dispositivos

- Código-fonte de um aplicativo de exemplo que mostra um exemplo de conexão MQTT em Python
- SDK de dispositivo AWS IoT para Python v2 no GitHub

JavaScript

Usar oAWS IoTSDK de dispositivo da para JavaScript Para conectar dispositivos

- Código-fonte de um aplicativo de exemplo que mostra um exemplo de conexão MQTT em JavaScript
- AWS IoTSDK de dispositivo da para JavaScript V2 no GitHub

Java

Usar oAWS IoTSDK de dispositivo para Java para conectar dispositivos

- Código-fonte de um aplicativo de exemplo que mostra um exemplo de conexão MQTT em Java
- SDK de dispositivo AWS IoT para Java v2 no GitHub

Embedded C

Usar oAWS IoTSDK de dispositivo da para C incorporado para conectar dispositivos

> **Important**
>
> Este SDK destina-se ao uso por desenvolvedores experientes de software incorporado.

- Código-fonte de um aplicativo de exemplo que mostra um exemplo de conexão MQTT em C incorporado
- SDK de dispositivo AWS IoT para C incorporado no GitHub

Opções de qualidade de serviço (QoS) MQTT

AWS IoTO e aAWS IoTOs SDKs do dispositivo oferecem suporte aoNíveis de qualidade de serviço (QoS) MQTT`0`e`1`. O protocolo MQTT define um terceiro nível de QoS, nível`2`, masAWS IoTO não oferece suporte a ele. Somente o protocolo MQTT suporta o recurso QoS. HTTPS suporta QoS passando um parâmetro de string de consulta`?qos=qos`onde o valor pode ser 0 ou 1.

Esta tabela descreve como cada nível de QoS afeta as mensagens publicadas para e pelo corretor de mensagens.

Com um nível de QoS de...	A mensagem é...	Comentários
QoS nível 0	Enviado zero ou mais vezes	Esse nível deve ser usado para mensagens enviadas por links de comunicação confiáveis ou que podem ser perdidas sem problemas.
QoS nível 1	Enviado pelo menos uma vez e, em seguida, repetidamente até um`PUBACK`resposta é recebida	A mensagem não é considerada completa até que o remetente receba um`PUBACK`resposta para indicar entrega bem-sucedida.

Usar sessões persistentes do MQTT

Sessões persistentes armazenam assinaturas e mensagens de um cliente, com uma qualidade de serviço (QoS) de 1, que não foram reconhecidas pelo cliente. Quando um dispositivo desconectado se reconecta a uma sessão persistente, a sessão é retomada, suas assinaturas são restabelecidas e as mensagens inscritas recebidas antes da reconexão e que não foram reconhecidas pelo cliente são enviadas ao cliente.

Criando uma sessão persistente

Você cria uma sessão persistente do MQTT ao enviar um`CONNECT`mensagem e configuração do`cleanSession`Sinalizador para`0`. Se nenhuma sessão existir para o cliente enviar a`CONNECT`mensagem, uma nova sessão persistente é criada. Se uma sessão já existir para o cliente, o cliente retomará a sessão existente.

Operações durante uma sessão persistente

Os clientes usam o`sessionPresent`atributo na conexão confirmada (`CONNACK`) mensagem para determinar se uma sessão persistente está presente. Se`sessionPresent`é`1`Uma sessão persistente está presente e quaisquer mensagens armazenadas para o cliente são entregues ao cliente imediatamente após o cliente receber o`CONNACK`, conforme descrito emTráfego de mensagens após a reconexão com uma sessão persistente (p. 89). Se`sessionPresent`é`1`, não há necessidade de o cliente reenviar. No entanto, se`sessionPresent`é`0`, nenhuma sessão persistente está presente e o cliente deve se inscrever novamente em seus filtros de tópico.

Depois que o cliente ingressa em uma sessão persistente, ele poderá publicar mensagens e assinar filtros de tópico sem nenhum sinalizador adicional em cada operação.

Tráfego de mensagens após a reconexão com uma sessão persistente

Uma sessão persistente representa uma conexão contínua entre um cliente e um agente de mensagens MQTT. Quando um cliente se conecta ao agente de mensagens da usando uma sessão persistente, o agente de mensagens salva todas as inscrições que o cliente faz durante a conexão. Quando o cliente se desconecta, o agente de mensagens armazena mensagens de QoS não confirmadas e novas mensagens de QoS 1 publicadas em tópicos nos quais o cliente está inscrito. As mensagens são armazenadas de acordo com o limite da conta, as mensagens que excedam esse limite serão descartadas. Para obter mais informações sobre limites de mensagens persistentes, consulteAWS IoT CoreEndpoints e cotas do. Quando o cliente se conecta novamente à sessão persistente, todas as inscrições são restabelecidas e todas as mensagens armazenadas são enviadas ao cliente a uma taxa máxima de 10 mensagens por segundo.

Após a reconexão, as mensagens armazenadas são enviadas para o cliente, a uma taxa limitada a 10 mensagens armazenadas por segundo, juntamente com qualquer tráfego de mensagens atual até o`Publish requests per second per connection`Limite é atingido. Como a taxa de entrega das mensagens armazenadas é limitada, levará vários segundos para entregar todas as mensagens armazenadas se uma sessão tiver mais de 10 mensagens armazenadas para serem entregues após a reconexão.

Encerrando uma sessão persistente

As seguintes condições descrevem como as sessões persistentes podem terminar.

- Quando o tempo de expiração da sessão persistente decorre. O temporizador de expiração de sessão persistente começa quando o agente de mensagens detecta que um cliente foi desconectado, seja pela desconexão do cliente ou pelo tempo limite da conexão.
- Quando o cliente envia um`CONNECT`mensagem que define o`cleanSession`Sinalizador para`1`.

> **Note**
>
> As mensagens armazenadas aguardando para serem enviadas ao cliente quando uma sessão termina são descartadas; no entanto, elas ainda são cobradas na taxa de mensagens padrão, mesmo que não pudessem ser enviadas. Para obter mais informações sobre definição de preço de mensagens, consulteAWS IoT CoreDefinição de preços. Você pode configurar o intervalo de tempo de expiração.

Reconexão após a expiração de uma sessão persistente

Se um cliente não se reconectar à sessão persistente antes de expirar, a sessão será encerrada e suas mensagens armazenadas serão descartadas. Quando um cliente se reconecta após a sessão expirar com um`cleanSession`Sinalizador para`0`, o serviço cria uma nova sessão persistente. Quaisquer assinaturas ou mensagens da sessão anterior não estão disponíveis para esta sessão porque foram descartadas quando a sessão anterior expirou.

Cobranças de mensagem de sessão persistente

As mensagens são cobradas em seuConta da AWSquando o corretor de mensagens envia uma mensagem para um cliente ou uma sessão persistente offline. Quando um dispositivo off-line com uma sessão persistente se reconecta e retoma sua sessão, as mensagens armazenadas são entregues ao dispositivo e cobradas em sua conta novamente. Para obter mais informações sobre definição de preço de mensagens, consulteAWS IoT Corepreços - Mensagens.

O tempo de expiração da sessão persistente padrão de uma hora pode ser aumentado usando o processo de aumento de limite padrão. Observe que aumentar o tempo de expiração da sessão pode aumentar

as cobranças de mensagens porque o tempo adicional pode permitir que mais mensagens sejam armazenadas para o dispositivo off-line e essas mensagens adicionais seriam cobradas na sua conta na taxa de mensagens padrão. O tempo de expiração da sessão é aproximado e uma sessão pode persistir por até 30 minutos a mais do que o limite da conta; no entanto, uma sessão não será menor que o limite da conta. Para obter mais informações sobre limites de sessão, consulteAWSCotas de serviço do.

Usar mensagens retidas do MQTT

AWS IoT Coresuporta o sinalizador RETAIN descrito emMQTT v3.1.1. Quando um cliente define o sinalizador RETAIN em uma mensagem MQTT publicada,AWS IoT Coresalva a mensagem. Ele pode então ser enviado para novos assinantes, recuperado chamando o`GetRetainedMessage`operação, e visualizado naAWS IoTconsole.AWS IoT Corearmazena mensagens retidas por três anos após a última vez em que foram atualizadas ou acessadas. Após três anos, as mensagens são excluídas.

Exemplos de uso de mensagens retidas do MQTT

- Como mensagem de configuração inicial

  As mensagens retidas da MQTT são enviadas a um cliente depois que o cliente assina um tópico. Se você quiser que todos os clientes que assinam um tópico recebam a mensagem mantida do MQTT imediatamente após a assinatura, você pode publicar uma mensagem de configuração com o sinalizador RETAIN definido. Os clientes assinantes também recebem atualizações dessa configuração sempre que uma nova mensagem de configuração é publicada.
- Como última mensagem de estado conhecida

  Os dispositivos podem definir o sinalizador RETAIN em mensagens de estado atual para queAWS IoT Coreos salvará. Quando os aplicativos se conectam ou se reconectam, eles podem se inscrever neste tópico e obter o último estado relatado imediatamente após se inscrever no tópico da mensagem retida. Dessa forma, eles podem evitar ter que esperar até a próxima mensagem do dispositivo para ver o estado atual.

Nesta seção:

- Tarefas comuns com mensagens retidas do MQTT emAWS IoT Core (p. 90)
- Mensagens de cobrança e retidas (p. 92)
- Comparando mensagens retidas do MQTT e sessões persistentes do MQTT (p. 92)
- MQTT manteve mensagens eAWS IoTSombras de dispositivos (p. 95)

Tarefas comuns com mensagens retidas do MQTT emAWS IoT Core

AWS IoT CoreSalva mensagens MQTT com o sinalizador RETAIN definido. EstesMensagens retidassão enviados para todos os clientes que se inscreveram no tópico, como uma mensagem MQTT normal, e eles também são armazenados para serem enviados para novos assinantes do tópico.

As mensagens retidas do MQTT exigem ações de política específicas para autorizar os clientes a acessá-las. Para obter exemplos de uso de políticas de mensagens retidas, consulteExemplos de política de mensagens retidas (p. 365).

Esta seção descreve operações comuns que envolvem mensagens retidas.

- Criando uma mensagem retida

  O cliente determina se uma mensagem é mantida quando publica uma mensagem MQTT. Os clientes podem definir o sinalizador RETAIN quando publicarem uma mensagem usando umSDK de dispositivo da (p. 1278). Aplicativos e serviços podem definir o sinalizador RETAIN quando eles usam o`Publishação`para publicar uma mensagem MQTT.

Apenas uma mensagem por nome de tópico é mantida. Uma nova mensagem com o sinalizador RETAIN publicado em um tópico substitui qualquer mensagem mantida existente enviada para o tópico anteriormente.

OBSERVAÇÃO: Você não pode publicar em umTópico reservado (p. 101)com o sinalizador RETAIN definido.

- Assinatura de um tópico de mensagem retida

  Os clientes assinam tópicos de mensagens retidas como fariam com qualquer outro tópico de mensagem MQTT. As mensagens retidas recebidas por assinatura de um tópico de mensagem retida têm o sinalizador RETAIN definido.

  As mensagens retidas são excluídas doAWS IoT Corequando um cliente publica uma mensagem retida com uma carga útil de mensagem de 0 bytes no tópico da mensagem retida. Os clientes que se inscreveram no tópico da mensagem retida também receberão a mensagem de 0 bytes.

  A assinatura de um filtro de tópicos curinga que inclui um tópico de mensagem retida permite que o cliente receba mensagens subsequentes publicadas no tópico da mensagem retida, mas não entrega a mensagem retida após a assinatura.

  OBSERVAÇÃO: Para receber uma mensagem retida após a assinatura, o filtro de tópico na solicitação de assinatura deve corresponder exatamente ao tópico da mensagem retida.

  As mensagens retidas recebidas ao assinar um tópico de mensagem retida têm o sinalizador RETAIN definido. Mensagens retidas recebidas por um cliente assinante após a assinatura, não.
- Recuperando uma mensagem retida

  As mensagens retidas são entregues aos clientes automaticamente quando eles se inscrevem no tópico com a mensagem retida. Para que um cliente receba a mensagem retida após a assinatura, ele deve se inscrever no nome exato do tópico da mensagem retida. Assinar um filtro de tópico curinga que inclui um tópico de mensagem retida permite que o cliente receba mensagens subsequentes publicadas no tópico da mensagem retida, mas não entrega a mensagem retida após a assinatura.

  Serviços e aplicativos podem listar e recuperar mensagens retidas chamando`ListRetainedMessages`e`GetRetainedMessage`.

  Um cliente não é impedido de publicar mensagens em um tópico de mensagem retidasemdefinindo o sinalizador RETAIN. Isso pode causar resultados inesperados, como a mensagem retida não corresponder à mensagem recebida assinando o tópico.
- Listando tópicos de mensagens retidas

  Você pode listar mensagens retidas chamando`ListRetainedMessages`e as mensagens retidas podem ser visualizadas naAWS IoTconsole.
- Obtendo detalhes da mensagem retida

  Você pode obter detalhes da mensagem retida ligando`GetRetainedMessage`e eles podem ser visualizados noAWS IoTconsole.
- Mantendo uma mensagem de testamento

  MQTTWillmensagensque são criados quando um dispositivo se conecta pode ser mantido definindo o`Will Retain`Sinalizador no`Connect Flag bits`campo.
- Excluir uma mensagem retida

  Dispositivos, aplicativos e serviços podem excluir uma mensagem retida publicando uma mensagem com o sinalizador RETAIN definido e uma carga de mensagem vazia (0 bytes) para o nome do tópico da mensagem retida a ser excluída. Essas mensagens excluem a mensagem retida deAWS IoT Core, são enviados para clientes com uma assinatura do tópico, mas eles não são retidos porAWS IoT Core.

As mensagens retidas também podem ser excluídas interativamente acessando a mensagem retida naAWS IoTconsole. Mensagens retidas que são excluídas usando oAWS IoTconsoletambém envia uma mensagem de 0 bytes para clientes que se inscreveram no tópico da mensagem retida.

As mensagens retidas não podem ser restauradas após serem excluídas. Um cliente precisaria publicar uma nova mensagem retida para tomar o lugar da mensagem excluída.

- Depuração e solução de problemas de mensagens retidas

  OAWS IoTconsoleFornece várias ferramentas para ajudá-lo a solucionar problemas de mensagens retidas:

  - OMensagens retidaspágina

    OMensagens retidasna página doAWS IoTconsole fornece uma lista paginada das mensagens retidas que foram armazenadas por sua Conta na Região atual. Nessa página, é possível:

    - Veja os detalhes de cada mensagem retida, como a carga útil da mensagem, QoS, a hora em que foi recebida.
    - Atualize o conteúdo de uma mensagem retida.
    - Exclua uma mensagem retida.
  - OCliente de teste MQTT

    OCliente de teste MQTTna página doAWS IoTO console pode assinar e publicar em tópicos do MQTT. A opção de publicação permite que você defina o sinalizador RETAIN nas mensagens publicadas para simular como seus dispositivos podem se comportar.

  Alguns resultados inesperados podem ser o resultado desses aspectos de como as mensagens retidas são implementadas emAWS IoT Core.

  - Limites de mensagens retidos

    Quando uma conta tiver armazenado o número máximo de mensagens retidas,AWS IoT Coreretorna uma resposta limitada às mensagens publicadas com o conjunto RETAIN e cargas úteis maiores que 0 bytes até que algumas mensagens retidas sejam excluídas e a contagem de mensagens retidas fique abaixo do limite.
  - Pedido de entrega de mensagem retida

    A sequência de mensagens retidas e entrega de mensagens inscritas não é garantida.

Mensagens de cobrança e retidas

Publicando mensagens com o sinalizador RETAIN definido a partir de um cliente, usandoAWS IoTconsole ou chamando`Publish`incorre em cobranças adicionais de mensagens descritas emAWS IoT Corepreços - Mensagens.

Recuperando mensagens retidas por um cliente, usandoAWS IoTconsole ou chamando`GetRetainedMessage`incorre em cobranças de mensagens além das cobranças normais de uso da API. As cobranças adicionais são descritas emAWS IoT Corepreços - Mensagens.

MQTTWillmensagensque são publicados quando um dispositivo se desconecta inesperadamente incorrem em cobranças de mensagens descritas emAWS IoT Corepreços - Mensagens.

Para obter mais informações sobre custos de mensagens, consulteAWS IoT Corepreços - Mensagens.

Comparando mensagens retidas do MQTT e sessões persistentes do MQTT

Mensagens retidas e sessões persistentes são recursos padrão do MQTT 3.1.1 que permitem que os dispositivos recebam mensagens publicadas enquanto estavam off-line. Mensagens retidas podem ser

publicadas a partir de sessões persistentes. Esta seção descreve os principais aspectos desses recursos e como eles funcionam em conjunto.

	Mensagens retidas	Sessões persistentes	Mensagens retidas em sessões persistentes
Recursos principais	As mensagens retidas podem ser usadas para configurar ou notificar grandes grupos de dispositivos depois que eles se conectam. Mensagens retidas também podem ser usadas onde você deseja que os dispositivos recebam somente a última mensagem publicada em um tópico após uma reconexão.	Sessões persistentes são úteis para dispositivos que têm conectividade intermitente e podem perder várias mensagens importantes. Os dispositivos podem se conectar a uma sessão persistente para receber mensagens enviadas enquanto estiverem off-line.	As mensagens retidas podem ser usadas em sessões regulares e persistentes.
Exemplos	As mensagens retidas podem fornecer aos dispositivos informações de configuração sobre seu ambiente quando eles ficam online. A configuração inicial pode incluir uma lista de outros tópicos de mensagens aos quais ela deve se inscrever ou informações sobre como ela deve configurar seu fuso horário local.	Os dispositivos que se conectam através de uma rede celular com conectividade intermitente podem usar sessões persistentes para evitar a falta de mensagens importantes que são enviadas enquanto um dispositivo está fora da cobertura da rede ou precisa desligar o rádio celular.	O dispositivo celular na amostra de sessão persistente pode usar uma mensagem retida para receber sua configuração inicial em sua conexão inicial.
Mensagens recebidas na assinatura inicial de um tópico	Depois de se inscrever em um tópico com uma mensagem retida, a mensagem retida mais recente é recebida.	Depois de se inscrever em um tópico sem uma mensagem retida, nenhuma mensagem é recebida até que uma seja publicada no tópico.	Depois de se inscrever em um tópico com uma mensagem retida, a mensagem retida mais recente é recebida.
Tópicos inscritos após a reconexão	Sem uma sessão persistente, o cliente deve se inscrever em tópicos após a reconexão.	Tópicos inscritos são restaurados após a reconexão.	Tópicos inscritos são restaurados após a reconexão.
Mensagens recebidas após a reconexão	Depois de se inscrever em um tópico com uma mensagem retida, a mensagem retida mais recente é recebida.	Todas as mensagens publicadas com um QOS = 1 e inscritas com um QOS =1 enquanto o dispositivo foi desconectado	Todas as mensagens publicadas com um QOS = 1 e inscritas com um QOS =1 que foram enviadas enquanto o dispositivo

	Mensagens retidas	Sessões persistentes	Mensagens retidas em sessões persistentes
		são enviadas após a reconexão do dispositivo.	foi desconectado são enviadas após a reconexão do dispositivo. Mensagens retidas atualizadas de tópicos nos quais o cliente foi inscrito também são enviadas ao cliente. Se mais de uma mensagem retida tiver sido publicada em um tópico enquanto o cliente estava offline, ele poderá receber várias mensagens armazenadas retidas para esse tópico depois de reconectar.
Validade de dados/ sessão	As mensagens retidas não expiram. Eles são armazenados até serem substituídos ou excluídos.	As sessões persistentes expiram se o cliente não se reconectar dentro do período de tempo limite. Depois que uma sessão persistente expira, as assinaturas do cliente e as mensagens salvas que foram publicadas com um QOS = 1 e inscritas com um QOS =1 enquanto o dispositivo foi desconectado são excluídas. Para obter mais informações sobre expirações de sessão com sessões persistentes, consultethe section called “Usar sessões persistentes do MQTT” (p. 88).	As mensagens retidas não expiram. Eles são armazenados até serem substituídos ou excluídos, mesmo que sejam enviados de dentro de uma sessão persistente que expirou. Depois que uma sessão persistente expira, as assinaturas do cliente e as mensagens salvas que foram publicadas com um QOS = 1 e inscritas com um QOS =1 enquanto o dispositivo foi desconectado são excluídas.

Para obter informações sobre sessões persistentes, consultethe section called “Usar sessões persistentes do MQTT” (p. 88).

Com as Mensagens retidas, o cliente de publicação determina se uma mensagem deve ser mantida e entregue a um dispositivo após a conexão, se ela tinha uma sessão anterior ou não. A opção de armazenar uma mensagem é feita pelo editor e a mensagem armazenada é entregue a todos os clientes

atuais e future que se inscrevem com assinaturas QoS 0 ou QoS 1. As mensagens retidas mantêm apenas uma mensagem em um determinado tópico por vez.

Quando uma conta tiver armazenado o número máximo de mensagens retidas,AWS IoT Coreretorna uma resposta limitada às mensagens publicadas com o conjunto RETAIN e cargas úteis maiores que 0 bytes até que algumas mensagens retidas sejam excluídas e a contagem de mensagens retidas fique abaixo do limite.

MQTT manteve mensagens eAWS IoTSombras de dispositivos

As mensagens retidas e as Sombras de Dispositivos retêm dados de um dispositivo, mas elas se comportam de forma diferente e servem a propósitos diferentes. Esta seção descreve suas semelhanças e diferenças.

	Mensagens retidas	Sombras de dispositivos
A carga útil da mensagem tem uma estrutura ou esquema predefinidos	Conforme definido pela implementação. O MQTT não especifica uma estrutura ou esquema para a carga útil da mensagem.	AWS IoTsuporta uma estrutura de dados específica.
Atualizar a carga útil da mensagem gera mensagens de evento	Publicar uma mensagem retida envia a mensagem para clientes inscritos, mas não gera mensagens de atualização adicionais.	A atualização de um Device Shadow produzatualizar mensagens que descrevem a alteração.
As atualizações de mensagens são numeradas	As mensagens retidas não são numeradas automaticamente.	Os documentos Device Shadow têm números de versão automáticos e carimbos de data/hora.
A carga útil da mensagem está anexada a um recurso de coisas	As mensagens retidas não são anexadas a um recurso de coisa.	Sombras de dispositivos são anexadas a um recurso de coisa.
Atualizando elementos individuais da carga útil da mensagem	Elementos individuais da mensagem não podem ser alterados sem atualizar toda a carga da mensagem.	Elementos individuais de um documento Device Shadow podem ser atualizados sem a necessidade de atualizar todo o documento Device Shadow.
O cliente recebe dados de mensagens após a assinatura	O cliente recebe automaticamente uma mensagem retida depois de assinar um tópico com uma mensagem retida.	Os clientes podem se inscrever nas atualizações do Device Shadow, mas devem solicitar o estado atual deliberadamente.
Indexação e pesquisa	As mensagens retidas não são indexadas para pesquisa.	Índices de indexação de frota dados Device Shadow para pesquisa e agregação.

Usar ConnectAttribut

`ConnectAttributes`permite especificar quais atributos você deseja usar em sua mensagem de conexão em suas políticas do IAM, como`PersistentConnecteLastWill`. com`ConnectAttributes`, você pode criar políticas que não dão aos dispositivos acesso a novos recursos por padrão, o que pode ser útil se um dispositivo estiver comprometido.

O `connectAttributes` oferece suporte aos seguintes recursos:

`PersistentConnect`

Usar a`PersistentConnect`Para salvar todas as inscrições que o cliente faz durante a conexão quando a conexão entre o cliente e o agente é interrompido.

`LastWill`

Usar a`LastWill`Para publicar uma mensagem no`LastWillTopic`quando um cliente se desconecta inesperadamente.

Por padrão, sua política tem conexão não persistente e não há atributos passados para essa conexão. Você deve especificar uma conexão persistente em sua política do IAM se quiser ter uma.

para o`ConnectAttributes`Exemplos, consulteExemplos de política de conexão (p. 343).

AWS IoTDiferenças da especificação MQTT versão 3.1.1

A implementação do corretor de mensagens é baseada naEspecificação MQTT v3.1.1, mas ela difere da especificação das seguintes maneiras:

- AWS IoTSuporta apenas os níveis 0 e 1 da MQTT da qualidade de serviço (QoS).AWS IoTnão suporta publicação ou assinatura com QoS nível 2. Quando o nível 2 da QoS é solicitado, o agente de mensagens não envia um PUBACK ou SUBACK.
- No AWS IoT, a assinatura de um tópico com nível 0 da QoS significa que uma mensagem é entregue zero ou mais vezes. Uma mensagem pode ser entregue mais de uma vez. As mensagens entregues mais de uma vez podem ser enviadas com outro ID de pacote. Nesses casos, o sinalizador DUP não é definido.
- Ao responder a uma solicitação de conexão, o operador de mensagens envia uma mensagem CONNACK. Essa mensagem contém um sinalizador para indicar se a conexão está retomando uma sessão anterior.
- Antes de enviar pacotes de controle adicionais ou uma solicitação de desconexão, o cliente deve aguardar que a mensagem CONNACK seja recebida em seu dispositivo a partir doAWS IoTAgente de mensagens.
- Quando um cliente se inscreve em um tópico, pode haver uma demora entre o momento em que o operador envia uma mensagem SUBACK e o momento em que o cliente começa a receber novas mensagens correspondentes.
- Quando um cliente usa o caractere curinga#No filtro de tópico da para assinar um tópico, todas as strings de caracteres em e abaixo de seu nível na hierarquia de tópicos são correspondidas. No entanto, o tópico pai não corresponde. Por exemplo, uma assinatura do tópico`sensor/#`recebe mensagens publicadas nos tópicos`sensor/`,`sensor/temperature`,`sensor/temperature/room1`, mas não mensagens publicadas para`sensor`. Para obter mais informações sobre curingas, consulteFiltros de tópicos (p. 100).
- O operador de mensagens usa o ID do cliente para identificar cada cliente. O ID do cliente é transmitido do cliente para o operador de mensagens como parte da carga útil do MQTT. Dois clientes com o mesma ID de cliente não podem ser conectados simultaneamente ao operador de mensagens. Quando um cliente se conecta ao agente de mensagens usando um ID que outro cliente está usando, a nova conexão do cliente é aceita e o cliente conectado anteriormente é desconectado.
- Em raras ocasiões, o operador de mensagens pode reenviar a mesma mensagem lógica PUBLISH com um ID de pacote diferente.
- A assinatura de filtros de tópicos que contêm um caractere curinga não pode receber mensagens retidas. Para receber uma mensagem retida, a solicitação de assinatura deve conter um filtro de tópico que corresponda exatamente ao tópico da mensagem retida.
- O agente de mensagens não garante a ordem em que as mensagens e o ACK são recebidos.

HTTPS

Os clientes podem publicar mensagens fazendo solicitações para a API REST usando os protocolos HTTP 1.0 ou 1.1. Para os mapeamentos de porta e autenticação usados por solicitações HTTP, consulte Protocolos, mapeamentos de porta e autenticação (p. 84).

> **Note**
>
> Ao contrário de MQTT, HTTPS não oferece suporte a um valor de `clientId`. Então, enquanto um`clientId`Está disponível ao usar MQTT, ele não está disponível ao usar HTTPS.

URL da mensagem HTTPS

Os dispositivos e clientes publicam suas mensagens fazendo solicitações POST para um endpoint específico do cliente e um URL específico do tópico:

```
https://IoT_data_endpoint/topics/url_encoded_topic_name?qos=1"
```

- *`IOT_DATA_Endpoint`*é oAWS IoTendpoint de dados do dispositivo (p. 80). Você pode encontrar o endpoint noAWS IoTConsole na página de detalhes da coisa ou no cliente usando aAWS CLIComando da :

  ```
  aws iot describe-endpoint --endpoint-type
                                      iot:Data-ATS
  ```

  O endpoint deve ser algo parecido com isto: `a3qjEXAMPLEffp-ats.iot.us-west-2.amazonaws.com`
- *`url_encoded_topic_name`* é o nome de tópico (p. 99) completo da mensagem que está sendo enviada.

Exemplos de código de mensagem HTTPS

Estes são alguns exemplos de como enviar uma mensagem HTTPS paraAWS IoT.

Python

```
import requests
import argparse

# define command-line parameters
parser = argparse.ArgumentParser(description="Send messages through an HTTPS
 connection.")
parser.add_argument('--endpoint', required=True, help="Your AWS IoT data custom
 endpoint, not including a port. " +
                                                        "Ex: \"abcdEXAMPLExyz-ats.iot.us-
east-1.amazonaws.com\"")
parser.add_argument('--cert', required=True, help="File path to your client
 certificate, in PEM format.")
parser.add_argument('--key', required=True, help="File path to your private key, in PEM
 format.")
parser.add_argument('--topic', required=True, default="test/topic", help="Topic to
 publish messages to.")
parser.add_argument('--message', default="Hello World!", help="Message to publish. " +
                                                        "Specify empty string to publish
 nothing.")

# parse and load command-line parameter values
args = parser.parse_args()
```

```
# create and format values for HTTPS request
publish_url = 'https://' + args.endpoint + ':8443/topics/' + args.topic + '?qos=1'
publish_msg = args.message.encode('utf-8')

# make request
publish = requests.request('POST',
            publish_url,
            data=publish_msg,
            cert=[args.cert, args.key])

# print results
print("Response status: ", str(publish.status_code))
if publish.status_code == 200:
        print("Response body:", publish.text)
```

CURL

Você pode usarcachode um cliente ou dispositivo para enviar uma mensagemAWS IoT.

Como usar curl a fim de enviar uma mensagem de um dispositivo de cliente do AWS IoT

1. Verifique ocurlVersão do .

   a. No cliente, execute esse comando em um prompt de comando.

      curl --help

      No texto de ajuda, procure as opções de TLS. Você deve ver a opção `--tlsv1.2`.

   b. Se você vir a opção `--tlsv1.2`, continue.

   c. Se você não vir o`--tlsv1.2`opção ou você obtém um`command not found`erro, talvez seja necessário atualizar ou instalar o curl em seu cliente ou instalar`openssl`Antes de continuar.

2. Instale os certificados no cliente.

   Copie os arquivos de certificado que você criou ao registrar o cliente (coisa) no console do AWS IoT. Verifique se você tem esses três arquivos de certificado no cliente antes de continuar.

   - O arquivo de certificado CA (*`Amazon-raiz-CA-1.pem`*neste exemplo).
   - O arquivo de certificado do cliente (*`device.pem.crt`*neste exemplo).
   - O arquivo de chave privada do cliente (*`private.pem.key`*neste exemplo).

3. Criação docurllinha de comando, substituindo os valores substituíveis para os da sua conta e do sistema.

```
curl --tlsv1.2 \
    --cacert Amazon-root-CA-1.pem \
    --cert device.pem.crt \
    --key private.pem.key \
    --request POST \
    --data "{ \"message\": \"Hello, world\" }" \
    "https://IoT_data_endpoint:8443/topics/topic?qos=1"
```

   --tlsv1.2

   Use TLS 1.2 (SSL).

   —cacert*`Amazon-raiz-CA-1.pem`*

   O nome e o caminho do arquivo, se necessário, do certificado CA para verificar o peer.

   --cert*`device.pem.crt`*

   O nome e o caminho do arquivo de certificado do cliente, se necessário.

—key*private.pem.key*

O nome e o caminho do arquivo de chave privada do cliente, se necessário.

—request POST

O tipo de solicitação HTTP (nesse caso, POST).

—data"*{\ "mensagem\": \ "Olá, mundo\"}*"

Os dados de HTTP POST que você deseja publicar. Nesse caso, é uma string JSON, com as aspas internas escapadas com o caractere de barra invertida (\).

"https://*IOT_DATA_Endpoint*:8443/tópicos/*tópico*? qos=1"

O URL do seu clienteAWS IoTendpoint de dados do dispositivo, seguido pela porta HTTPS,`:8443`, que é seguida pela palavra-chave,`/topics/`e o nome do tópico,`topic`Nesse caso,. Especifique a Qualidade de serviço como o parâmetro de consulta,`?qos=1`.

4. Abra o cliente de teste MQTT noAWS IoTconsole do .

   Siga as instruções emVisualizar mensagens MQTT com o cliente MQTT do AWS IoT (p. 66)E configure o console para assinar mensagens com o nome do tópico do*tópico*Usado em seucurl, ou use o filtro de tópico curinga do`#`.

5. Teste o comando.

   Ao monitorar o tópico no cliente de teste do console do AWS IoT, acesse o cliente e emita a linha de comando curl criada na etapa 3. Você deve ver as mensagens do cliente no console.

Tópicos MQTT

Tópicos MQTT identificamAWS IoTmensagens.AWS IoTOs cliente identificam as mensagens que publicam, dando nomes de tópicos às mensagens. Os clientes identificam as mensagens que desejam assinar (receber) registrando um filtro de tópico com o AWS IoT Core. O agente de mensagens do usa nomes de tópicos e filtros de tópicos para rotear mensagens de clientes de publicação a clientes assinatura.

O corretor de mensagens usa tópicos para identificar mensagens enviadas usando MQTT e enviadas usando HTTP para oURL da mensagem HTTPS (p. 97).

Embora o AWS IoT seja compatível com alguns tópicos reservados do sistema (p. 101), a maioria dos tópicos MQTT é criada e gerenciada por você, o designer do sistema. O AWS IoT usa tópicos para identificar mensagens recebidas de clientes de publicação e selecionar mensagens a serem enviadas para clientes assinantes, conforme descrito nas seções a seguir. Antes de criar um namespace de tópico para seu sistema, revise as características dos tópicos MQTT para criar a hierarquia de nomes de tópicos que funciona melhor para seu sistema de IoT.

Nomes de tópicos

Os nomes de tópicos e os filtros de tópicos são strings codificadas em UTF-8. Eles podem representar uma hierarquia de informações usando o caractere barra (/) para separar os níveis da hierarquia. Por exemplo, o nome deste tópico pode se referir a um sensor de temperatura no cômodo 1:

- `sensor/temperature/room1`

Neste exemplo, também pode haver outros tipos de sensores em outros cômodos com nomes de tópicos, como:

- `sensor/temperature/room2`
- `sensor/humidity/room1`
- `sensor/humidity/room2`

> **Note**
>
> Ao considerar os nomes de tópicos para as mensagens no sistema, lembre-se:
>
> - Os nomes de tópicos e os filtros de tópico diferenciam letras maiúsculas de minúsculas.
> - Os nomes de tópicos não devem conter informações de identificação pessoal.
> - Os nomes de tópicos que começam com $ são tópicos reservados (p. 101) para serem usados somente pelo AWS IoT Core.
> - AWS IoT CoreO não pode enviar ou receber mensagens entreConta da AWSs ou Regiões.

Para obter mais informações sobre como criar nomes de tópicos e namespace, consulte nosso whitepaper,Projetando Tópicos MQTT paraAWS IoT Core.

Para exemplos de como os aplicativos podem publicar e assinar mensagens, comece comConceitos básicos do AWS IoT Core (p. 18)eAWS IoTSDKs de dispositivos, SDKs móveis eAWS IoTCliente de dispositivo (p. 1278).

> **Important**
>
> O namespace de tópico é limitado a umConta da AWSe Região. Por exemplo, as receitas`sensor/temp/room1`tópico usado por umConta da AWSEm uma região é diferente da`sensor/temp/room1`tópico usado pelo mesmoAWSconta em outra região ou usada por qualquer outraConta da AWSem qualquer região da.

Topic ARN

Todos os ARNs de tópicos (Amazon Resource Names) têm o seguinte formulário:

```
arn:aws:iot:aws-region:AWS-account-ID:topic/Topic
```

Por exemplo,`arn:aws:iot:us-west-2:123EXAMPLE456:topic/application/topic/device/sensor`é um ARN para o tópico `application/topic/device/sensor`.

Filtros de tópicos

Os clientes assinantes registram filtros de tópico com o agente de mensagens do para especificar os tópicos de mensagem que o agente de mensagens deve enviar para eles. Um filtro de tópico pode ser o nome de um tópico único para assinar o nome de um tópico único ou pode incluir caracteres curinga para assinar vários nomes de tópicos ao mesmo tempo.

Os cliente de publicação não podem usar caracteres curinga nos nomes de tópicos que publicam.

A tabela a seguir lista os caracteres curinga que podem ser usados em um filtro de tópico.

Curingas de tópicos

Caractere curinga	Correspondências	Observações
#	Todas as strings de caracteres em e abaixo de seu nível na hierarquia de tópicos.	Deve ser o último caractere no filtro de tópico. Deve ser o único caractere no nível da hierarquia de tópicos.

Caractere curinga	Correspondências	Observações
		Pode ser usado em um filtro de tópico que também contém o caractere curinga +.
+	Qualquer string no nível que contém o caractere.	Deve ser o único caractere no nível da hierarquia de tópicos. Pode ser usado em vários níveis de um filtro de tópico.

Exemplos de uso de curingas com o nome de tópico do sensor anterior:

- Uma assinatura de `sensor/#` recebe mensagens publicadas em `sensor/`, `sensor/temperature`, `sensor/temperature/room1`, mas não mensagens publicadas em `Sensor`.
- Uma assinatura de `sensor/+/room1` recebe mensagens publicadas em `sensor/temperature/room1` e `sensor/humidity/room1`, mas não mensagens enviadas para `sensor/temperature/room2` ou `sensor/humidity/room2`.

ARN do filtro do tópico

Todos os ARNs de filtro de tópicos (Amazon Resource Names) têm o seguinte formulário:

```
arn:aws:iot:aws-region:AWS-account-ID:topicfilter/TopicFilter
```

Por exemplo,`arn:aws:iot:us-west-2:123EXAMPLE456:topicfilter/application/topic/+/sensor`É um ARN para o filtro de tópicos `application/topic/+/sensor`.

Carga útil da mensagem MQTT

A carga útil da mensagem enviada em suas mensagens MQTT não é especificada porAWS IoT, a menos que seja para um dosthe section called “Tópicos reservados” (p. 101). Para atender às necessidades do aplicativo, recomendamos que você defina a carga útil da mensagem para seus tópicos dentro das restrições doAWS IoT CoreService Quotas para protocolos.

O uso de um formato JSON para a carga útil da mensagem habilita oAWS IoTmecanismo de regras para analisar suas mensagens e aplicar consultas SQL a elas. Se o aplicativo não exigir que o mecanismo de regras aplique consultas SQL às cargas úteis de mensagens, você poderá usar qualquer formato de dados necessário pelo aplicativo. Para obter informações sobre limitações e caracteres reservados em um documento JSON usado em consultas SQL, consulteExtensões JSON (p. 626).

Para obter mais informações sobre como projetar seus tópicos do MQTT e suas cargas de mensagens correspondentes, consulteProjetando Tópicos MQTT paraAWS IoT Core.

Se um limite de tamanho de mensagem exceder as cotas de serviço, isso resultará em um`CLIENT_ERROR`com motivo`PAYLOAD_LIMIT_EXCEEDED`e “A carga útil da mensagem excede o limite de tamanho para o tipo de mensagem.” Para obter mais informações sobre o limite de tamanho da mensagem, consulteAWS IoT Corelimites e cotas do corretor de mensagens.

Tópicos reservados

Os tópicos que começam com um cifrão ($) são reservados para uso pelo AWS IoT. É possível se inscrever e publicar nesses tópicos reservados conforme permitido. No entanto, não é possível criar tópicos que comecem com um cifrão. Operações de publicação ou assinatura sem suporte em tópicos reservados podem resultar no encerramento de uma conexão.

Tópicos de modelos de ativos

Tópico	Operações do cliente permitidas	Descrição
$aws/sitewise/asset-models/*assetModelId*/assets/*assetId*/properties/*propertyId*	Inscrever-se	O AWS IoT SiteWise publica notificações de propriedade de ativo neste tópico. Para obter mais informações, consulteInteragir com outrosAWSserviçosnoAWS IoT SiteWiseGuia do usuário do.

Tópicos do Device Defender

Essas mensagens oferecem suporte a buffers de resposta no formato CBOR (Concise Binary Object Representation) e JavaScript JSON (Object Notation), dependendo da*formato da carga*do tópico.

formato da carga	Tipo de dados do formato de resposta
cbor	Concise Binary Object Representation (CBOR)
json	JavaScript Object Notation (JSON)

Para obter mais informações, consulte Envio de métricas de dispositivos (p. 1027).

Tópico	Operações permitidas	Descrição
$aws/things/*thingName*/defender/metrics/*payload-format*	Publicar	Os agentes do Device Defender publicam métricas neste tópico. Para obter mais informações, consulte Envio de métricas de dispositivos (p. 1027).
$aws/things/*thingName*/defender/metrics/*payload-format*/accepted	Inscrever-se	O AWS IoT publica neste tópico depois que um agente do Device Defender publicar uma mensagem bem-sucedida em $aws/things/*thingName*/defender/metrics/*payload-format*. Para obter mais informações, consulte Envio de métricas de dispositivos (p. 1027).
$aws/things/*thingName*/defender/metrics/*payload-format*/rejected	Inscrever-se	O AWS IoT publica neste tópico depois que um agente do Device Defender publicar uma mensagem malsucedida em $aws/things/*thingName*/defender/metrics/*payload-format*.

Tópico	Operações permitidas	Descrição
		Para obter mais informações, consulte Envio de métricas de dispositivos (p. 1027).

Tópicos de eventos

Tópico	Operações do cliente permitidas	Descrição
$aws/events/certificates/registered/*caCertificateId*	Inscrever-se	O AWS IoT publica esta mensagem quando o AWS IoT registra automaticamente um certificado e quando um cliente apresenta um certificado com o status `PENDING_ACTIVATION`. Para obter mais informações, consulte the section called "Configurar a primeira conexão feita por um cliente para registro automático" (p. 311).
$aws/events/job/*JobId*/cancelado	Inscrever-se	AWS IoTO publica essa mensagem quando um trabalho é cancelado. Para obter mais informações, consulte Eventos de trabalho (p. 1118).
$aws/events/job/*JobId*/cancellation_in_progress	Inscrever-se	AWS IoTpublica esta mensagem quando um trabalho está sendo cancelado. Para obter mais informações, consulte Eventos de trabalho (p. 1118).
$aws/events/job/*JobId*/complete	Inscrever-se	AWS IoTpublica esta mensagem quando um trabalho for concluído. Para obter mais informações, consulte Eventos de trabalho (p. 1118).
$aws/events/job/*JobId*/excluído	Inscrever-se	AWS IoTÉ publicada essa mensagem quando um trabalho é excluído Para obter mais informações, consulteEventos de trabalho (p. 1118).
$aws/events/job/*JobId*/deletion_in_progress	Inscrever-se	AWS IoTpublica essa mensagem quando um trabalho está sendo excluído. Para obter mais informações,

Tópico	Operações do cliente permitidas	Descrição
		consulte Eventos de trabalho (p. 1118).
$aws/events/jobExecution/*JobId*/cancelado	Inscrever-se	AWS IoTO publica essa mensagem quando a execução de uma tarefa é cancelada. Para obter mais informações, consulte Eventos de trabalho (p. 1118).
$aws/events/jobExecution/*JobId*/excluído	Inscrever-se	AWS IoTO publica essa mensagem quando a execução de uma tarefa é excluída. Para obter mais informações, consulte Eventos de trabalho (p. 1118).
$aws/events/jobExecution/*JobId*/failed	Inscrever-se	AWS IoTpublica essa mensagem quando uma execução de trabalho falhou. Para obter mais informações, consulte Eventos de trabalho (p. 1118).
$aws/events/jobExecution/*JobId*/rejected	Inscrever-se	AWS IoTpublica essa mensagem quando uma execução de trabalho foi rejeitada. Para obter mais informações, consulte Eventos de trabalho (p. 1118).
$aws/events/jobExecution/*JobId*/removed	Inscrever-se	AWS IoTpublica essa mensagem quando uma execução de trabalho foi removida. Para obter mais informações, consulte Eventos de trabalho (p. 1118).
$aws/events/jobExecution/*JobId*/sucedido	Inscrever-se	AWS IoTpublica essa mensagem quando uma execução de trabalho foi bem-sucedida. Para obter mais informações, consulte Eventos de trabalho (p. 1118).
$aws/events/jobExecution/*JobId*/timed_out	Inscrever-se	AWS IoTpublica essa mensagem quando uma execução de trabalho expirou. Para obter mais informações, consulte Eventos de trabalho (p. 1118).

Tópico	Operações do cliente permitidas	Descrição
$aws/events/presence/connected/*clientId*	Inscrever-se	A AWS IoT faz publicações nesse tópico quando um cliente MQTT com o ID do cliente especificado se conecta à AWS IoT. Para obter mais informações, consulte Eventos de conexão/desconexão (p. 1121).
$aws/events/presence/disconnected/*clientId*	Inscrever-se	A AWS IoT faz publicações nesse tópico quando um cliente MQTT com o ID do cliente especificado se desconecta à AWS IoT. Para obter mais informações, consulte Eventos de conexão/desconexão (p. 1121).
$aws/events/subscriptions/subscribed/*clientId*	Inscrever-se	A AWS IoT faz publicações nesse tópico quando um cliente MQTT com o ID do cliente especificado se inscreve em um tópico MQTT. Para obter mais informações, consulte Eventos de assinatura/cancelamento de assinatura (p. 1124).
$aws/events/subscriptions/unsubscribed/*clientId*	Inscrever-se	A AWS IoT faz publicações nesse tópico quando um cliente MQTT com o ID do cliente especificado cancela a inscrição em um tópico MQTT. Para obter mais informações, consulte Eventos de assinatura/cancelamento de assinatura (p. 1124).
$aws/events/thing/*thingName*/created	Inscrever-se	O AWS IoT publica neste tópico quando a coisa *thingName* é criada. Para obter mais informações, consulte the section called “Eventos de registro” (p. 1110).
$aws/events/thing/*thingName*/updated	Inscrever-se	O AWS IoT publica neste tópico quando a coisa *thingName* é atualizada. Para obter mais informações, consulte the section called “Eventos de registro” (p. 1110).

Tópico	Operações do cliente permitidas	Descrição
$aws/events/thing/*thingName*/deleted	Inscrever-se	O AWS IoT publica neste tópico quando a coisa *thingName* é excluída. Para obter mais informações, consulte the section called "Eventos de registro" (p. 1110).
$aws/events/thingGroup/*thingGroupName*/created	Inscrever-se	O AWS IoT publica neste tópico quando o grupo de coisas, *thingGroupName*, é criado. Para obter mais informações, consulte the section called "Eventos de registro" (p. 1110).
$aws/events/thingGroup/*thingGroupName*/updated	Inscrever-se	O AWS IoT publica neste tópico quando o grupo de coisas, *thingGroupName*, é atualizado. Para obter mais informações, consulte the section called "Eventos de registro" (p. 1110).
$aws/events/thingGroup/*thingGroupName*/deleted	Inscrever-se	O AWS IoT publica neste tópico quando o grupo de coisas, *thingGroupName*, é excluído. Para obter mais informações, consulte the section called "Eventos de registro" (p. 1110).
$aws/events/thingType/*thingTypeName*/created	Inscrever-se	O AWS IoT publica neste tópico quando o tipo de coisa *thingTypeName* é criado. Para obter mais informações, consulte the section called "Eventos de registro" (p. 1110).
$aws/events/thingType/*thingTypeName*/updated	Inscrever-se	O AWS IoT publica neste tópico quando o tipo de coisa *thingTypeName* é atualizado. Para obter mais informações, consulte the section called "Eventos de registro" (p. 1110).
$aws/events/thingType/*thingTypeName*/deleted	Inscrever-se	O AWS IoT publica neste tópico quando o tipo de coisa *thingTypeName* é excluído. Para obter mais informações, consulte the section called "Eventos de registro" (p. 1110).

Tópico	Operações do cliente permitidas	Descrição
$aws/events/thingTypeAssociation/thing/*thingName*/*thingTypeName*	Inscrever-se	O AWS IoT publica neste tópico quando a coisa, *thingName*, está associada ou desassociada do tipo de coisa, *thingTypeName*. Para obter mais informações, consulte the section called "Eventos de registro" (p. 1110).
$aws/events/thingGroupMembership/thingGroup/*thingGroupName*/thing/*thingName*/added	Inscrever-se	O AWS IoT publica neste tópico quando a coisa, *thingName*, é adicionada ao grupo de coisas, *thingGroupName*. Para obter mais informações, consulte the section called "Eventos de registro" (p. 1110).
$aws/events/thingGroupMembership/thingGroup/*thingGroupName*/thing/*thingName*/removed	Inscrever-se	O AWS IoT publica neste tópico quando a coisa, *thingName*, é removida do grupo de coisas, *thingGroupName*. Para obter mais informações, consulte the section called "Eventos de registro" (p. 1110).
$aws/events/thingGroupHierarchy/thingGroup/*parentThingGroupName*/childThingGroup/*childThingGroupName*/added	Inscrever-se	O AWS IoT publica neste tópico quando o grupo de coisas, *childThingGroupName*, é adicionado ao grupo de coisas, *parentThingGroupName*. Para obter mais informações, consulte the section called "Eventos de registro" (p. 1110).
$aws/events/thingGroupHierarchy/thingGroup/*parentThingGroupName*/childThingGroup/*childThingGroupName*/removed	Inscrever-se	O AWS IoT publica neste tópico quando o grupo de coisas, *childThingGroupName*, é removido do grupo de coisas, *parentThingGroupName*. Para obter mais informações, consulte the section called "Eventos de registro" (p. 1110).

Tópicos de provisionamento de frota

Essas mensagens oferecem suporte a buffers de resposta no formato CBOR (Concise Binary Object Representation) e JavaScript JSON (Object Notation), dependendo da*formato da carga*do tópico.

formato da carga	Tipo de dados do formato de resposta
cbor	Concise Binary Object Representation (CBOR)
json	JavaScript Object Notation (JSON)

Para obter mais informações, consulte API MQTT de provisionamento de dispositivos (p. 830).

Tópico	Operações do cliente permitidas	Descrição
$aws/certificates/create/*formato da carga*	Publicar	Publique nesse tópico para criar um certificado usando uma solicitação de assinatura de certificado (CSR).
$aws/certificates/create/*formato da carga*/accepted	Inscrever-se	O AWS IoT publica nesse tópico após uma chamada bem-sucedida para $aws/certificates/create/*formato da carga*.
$aws/certificates/create/*formato da carga*/rejected	Inscrever-se	AWS IoT publica nesse tópico após uma chamada malsucedida para $aws/certificates/create/*formato da carga*.
$aws/certificates/create-from-csr/*formato da carga*	Publicar	Publica nesse tópico para criar um certificado a partir de uma CSR.
$aws/certificates/create-from-csr/*formato da carga*/accepted	Inscrever-se	AWS IoT publica nesse tópico uma chamada bem-sucedida para $aws/certificates/create-from-csr/*formato da carga*.
$aws/certificates/create-from-csr/*formato da carga*/rejected	Inscrever-se	AWS IoT publica nesse tópico uma chamada malsucedida para $aws/certificates/create-from-csr/*formato da carga*.
$aws/events/presence/connected/*clientId*	Inscrever-se	A AWS IoT faz publicações nesse tópico quando um cliente MQTT com o ID do cliente especificado se conecta à AWS IoT. Para obter mais informações, consulte Eventos de conexão/desconexão (p. 1121).
$aws/provisioning-templates/*templateName*/provision/*formato da carga*	Publicar	Publique nesse tópico para registrar uma coisa.

Tópico	Operações do cliente permitidas	Descrição
$aws/provisioning-templates/*templateName*/provision/*formato da carga*/accepted	Inscrever-se	AWS IoT publica nesse tópico após uma chamada bem-sucedida para $aws/provisioning-templates/*templateName*/provision/*formato da carga*.
$aws/provisioning-templates/*templateName*/provision/*formato da carga*/rejected	Inscrever-se	AWS IoT publica nesse tópico após uma chamada malsucedida para $aws/provisioning-templates/*templateName*/provision/*formato da carga*.

Tópicos de trabalhos

Note

As operações do cliente observadas comoRecebernesta tabela indicam tópicos queAWS IoTpublica diretamente no cliente que o solicitou, independentemente de o cliente se inscrever no tópico ou não. Os clientes devem esperar receber essas mensagens de resposta, mesmo que não tenham se inscrito nelas.
Essas mensagens de resposta não passam pelo corretor de mensagens e não podem ser inscritas por outros clientes ou regras. Para assinar mensagens relacionadas à atividade de trabalho, use `onotifyenotify-next`Tópicos.
Ao se inscrever no trabalho `ejobExecution`tópicos de eventos para sua solução de monitoramento de frotas, você deve primeiro ativarJobIDs (p. 1107)para receber eventos no lado da nuvem.
Para obter mais informações, consulte API MQTT de dispositivos de trabalho (p. 756).

Tópico	Operações do cliente permitidas	Descrição
$aws/things/*thingName*/jobs/get	Publicar	Os dispositivos publicam uma mensagem neste tópico para fazer uma solicitação `GetPendingJobExecutions`. Para obter mais informações, consulte API MQTT de dispositivos de trabalho (p. 756).
$aws/things/*thingName*/jobs/get/accepted	Assine, receba	Os dispositivos se inscrevem nesse tópico para receber respostas bem-sucedidas de uma solicitação `GetPendingJobExecutions`. Para obter mais informações, consulte API MQTT de dispositivos de trabalho (p. 756).
$aws/things/*thingName*/jobs/get/rejected	Assine, receba	Os dispositivos se inscrevem nesse tópico

Tópico	Operações do cliente permitidas	Descrição
		para receber uma resposta quando`GetPendingJobExecutions`A solicitação foi rejeitada. Para obter mais informações, consulte API MQTT de dispositivos de trabalho (p. 756).
$aws/things/*`thingName`*/jobs/start-next	Publicar	Os dispositivos publicam uma mensagem neste tópico para fazer uma solicitação `StartNextPendingJobExecution`. Para obter mais informações, consulte API MQTT de dispositivos de trabalho (p. 756).
$aws/things/*`thingName`*/jobs/start-next/accepted	Assine, receba	Os dispositivos se inscrevem neste tópico para receber respostas bem-sucedidas para uma solicitação `StartNextPendingJobExecution`. Para obter mais informações, consulte API MQTT de dispositivos de trabalho (p. 756).
$aws/things/*`thingName`*/jobs/start-next/rejected	Assine, receba	Os dispositivos se inscrevem nesse tópico para receber uma resposta quando`StartNextPendingJobExecution` solicitação foi rejeitada. Para obter mais informações, consulte API MQTT de dispositivos de trabalho (p. 756).
$aws/things/*`thingName`*/jobs/*`jobId`*/get	Publicar	Os dispositivos publicam uma mensagem neste tópico para fazer uma solicitação `DescribeJobExecution`. Para obter mais informações, consulte API MQTT de dispositivos de trabalho (p. 756).
$aws/things/*`thingName`*/jobs/*`jobId`*/get/accepted	Assine, receba	Os dispositivos se inscrevem neste tópico para receber respostas bem-sucedidas para uma solicitação `DescribeJobExecution`. Para obter mais informações, consulte API MQTT de dispositivos de trabalho (p. 756).

Tópico	Operações do cliente permitidas	Descrição
$aws/things/*thingName*/jobs/*jobId*/get/rejected	Assine, receba	Os dispositivos se inscrevem nesse tópico para receber uma resposta quando`DescribeJobExecution`A solicitação foi rejeitada. Para obter mais informações, consulte API MQTT de dispositivos de trabalho (p. 756).
$aws/things/*thingName*/jobs/*jobId*/update	Publicar	Os dispositivos publicam uma mensagem neste tópico para fazer uma solicitação `UpdateJobExecution`. Para obter mais informações, consulte API MQTT de dispositivos de trabalho (p. 756).
$aws/things/*thingName*/jobs/*jobId*/update/accepted	Assine, receba	Os dispositivos se inscrevem neste tópico para receber respostas de êxito de uma solicitação `UpdateJobExecution`. Para obter mais informações, consulte API MQTT de dispositivos de trabalho (p. 756). Observação Somente o dispositivo que publica em $aws/things/*thingName*/jobs/*jobId*/update recebe mensagens sobre esse tópico.
$aws/things/*thingName*/jobs/*jobId*/update/rejected	Assine, receba	Os dispositivos se inscrevem nesse tópico para receber uma resposta quando`UpdateJobExecution`A solicitação foi rejeitada. Para obter mais informações, consulte API MQTT de dispositivos de trabalho (p. 756). Observação Somente o dispositivo que publica em $aws/things/*thingName*/jobs/*jobId*/update recebe mensagens sobre esse tópico.

Tópico	Operações do cliente permitidas	Descrição
$aws/things/*thingName*/jobs/notify	Inscrever-se	Os dispositivos se inscrevem neste tópico para receber notificações quando uma execução de tarefa é adicionada à lista de execuções pendentes de uma coisa ou removida dela. Para obter mais informações, consulte API MQTT de dispositivos de trabalho (p. 756).
$aws/things/*thingName*/jobs/notify-next	Inscrever-se	Os dispositivos se inscrevem neste tópico para receber notificações quando a próxima execução de tarefa pendente para a coisa é alterada. Para obter mais informações, consulte API MQTT de dispositivos de trabalho (p. 756).
$aws/events/job/*jobId*/completed	Inscrever-se	O serviço Jobs publica um evento neste tópico quando uma tarefa é concluída. Para obter mais informações, consulte Eventos de trabalho (p. 1118).
$aws/events/job/*jobId*/canceled	Inscrever-se	O serviço Jobs publica um evento neste tópico quando uma tarefa é cancelada. Para obter mais informações, consulte Eventos de trabalho (p. 1118).
$aws/events/job/*jobId*/deleted	Inscrever-se	O serviço Jobs publica um evento neste tópico quando uma tarefa é excluída. Para obter mais informações, consulte Eventos de trabalho (p. 1118).
$aws/events/job/*jobId*/cancellation_in_progress	Inscrever-se	O serviço Jobs publica um evento neste tópico quando o cancelamento de uma tarefa começa. Para obter mais informações, consulte Eventos de trabalho (p. 1118).
$aws/events/job/*jobId*/deletion_in_progress	Inscrever-se	O serviço Jobs publica um evento neste tópico quando a exclusão de tarefa começa. Para obter mais informações, consulte Eventos de trabalho (p. 1118).

Tópico	Operações do cliente permitidas	Descrição
$aws/events/jobExecution/*jobId*/succeeded	Inscrever-se	O serviço Jobs publica um evento neste tópico quando a execução da tarefa é bem-sucedida. Para obter mais informações, consulte Eventos de trabalho (p. 1118).
$aws/events/jobExecution/*jobId*/failed	Inscrever-se	O serviço Jobs publica um evento neste tópico quando uma execução de tarefa falha. Para obter mais informações, consulte Eventos de trabalho (p. 1118).
$aws/events/jobExecution/*jobId*/rejected	Inscrever-se	O serviço Jobs publica um evento neste tópico quando uma execução de tarefa é rejeitada. Para obter mais informações, consulte Eventos de trabalho (p. 1118).
$aws/events/jobExecution/*jobId*/canceled	Inscrever-se	O serviço Jobs publica um evento neste tópico quando a execução de uma tarefa é cancelada. Para obter mais informações, consulte Eventos de trabalho (p. 1118).
$aws/events/jobExecution/*jobId*/timed_out	Inscrever-se	O serviço Jobs publica um evento neste tópico quando a execução de uma tarefa atinge o tempo limite. Para obter mais informações, consulte Eventos de trabalho (p. 1118).
$aws/events/jobExecution/*jobId*/removed	Inscrever-se	O serviço Jobs publica um evento neste tópico quando a execução de uma tarefa é removida. Para obter mais informações, consulte Eventos de trabalho (p. 1118).
$aws/events/jobExecution/*jobId*/deleted	Inscrever-se	O serviço Jobs publica um evento neste tópico quando a execução de uma tarefa é excluída. Para obter mais informações, consulte Eventos de trabalho (p. 1118).

Tópicos de regras

Tópico	Operações do cliente permitidas	Descrição
$aws/rules/*ruleName*	Publicar	Um dispositivo ou um aplicativo faz publicações

Tópico	Operações do cliente permitidas	Descrição
		nesse tópico para acionar regras diretamente. Para obter mais informações, consulte Reduzir custos do sistema de mensagens com a Ingestão básica (p. 567).

Proteger tópicos de túneis

Tópico	Operações do cliente permitidas	Descrição
$aws/things/*thing-name*/tunnels/notify	Inscrever-se	O AWS IoT publica esta mensagem para que um agente de IoT inicie um proxy local no dispositivo remoto. Para obter mais informações, consulte the section called "Snippet de agente de IoT" (p. 795).

Tópicos de sombra

Os tópicos desta seção são usados por sombras nomeadas e sem nome. Os tópicos usados por cada uma diferem apenas no prefixo do tópico. Esta tabela mostra o prefixo do tópico usado em cada tipo de sombra.

Valor de ***ShadowTopicPrefix***	Tipo de sombra
$aws/things/*thingName*/shadow	Sombra sem nome (clássica)
$aws/things/*thingName*/shadow/name/*shadowName*	Sombra nomeada

Para criar um tópico completo, selecione o*ShadowTopicPrefix*para o tipo de sombra ao qual você deseja se referir, substitua*thingName*e, se aplicável,*shadowName*, com os valores correspondentes e acrescente isso ao stub de tópico, conforme mostrado na seguinte tabela. Lembre-se de que os tópicos diferenciam maiúsculas de minúsculas.

Tópico	Operações do cliente permitidas	Descrição
ShadowTopicPrefix/delete	Publicar/assinar	Um dispositivo ou um aplicativo faz publicações nesse tópico para excluir uma shadow Para obter mais informações, consulte /delete (p. 669).
ShadowTopicPrefix/delete/accepted	Inscrever-se	O serviço Device Shadow envia mensagens para esse tópico quando uma shadow é excluída. Para obter mais

Tópico	Operações do cliente permitidas	Descrição
		informações, consulte /delete/accepted (p. 670).
ShadowTopicPrefix/delete/rejected	Inscrever-se	O serviço Device Shadow envia mensagens para esse tópico quando uma solicitação de exclusão de uma shadow é rejeitada. Para obter mais informações, consulte /delete/rejected (p. 671).
ShadowTopicPrefix/get	Publicar/assinar	Um aplicativo ou uma coisa publica uma mensagem vazia nesse tópico para obter um shadow. Para obter mais informações, consulte Tópicos MQTT do Device Shadow (p. 663).
ShadowTopicPrefix/get/accepted	Inscrever-se	O serviço Device Shadow envia mensagens para esse tópico quando uma solicitação para uma shadow é feita com êxito. Para obter mais informações, consulte /get/accepted (p. 665).
ShadowTopicPrefix/get/rejected	Inscrever-se	O serviço Device Shadow envia mensagens para esse tópico quando uma solicitação de shadow é rejeitada. Para obter mais informações, consulte /get/rejected (p. 665).
ShadowTopicPrefix/update	Publicar/assinar	Uma coisa ou um aplicativo faz publicações nesse tópico para atualizar um shadow. Para obter mais informações, consulte /update (p. 666).
ShadowTopicPrefix/update/accepted	Inscrever-se	O serviço Device Shadow envia mensagens para esse tópico quando uma atualização é feita com êxito em uma shadow. Para obter mais informações, consulte /update/accepted (p. 668).
ShadowTopicPrefix/update/rejected	Inscrever-se	O serviço Device Shadow envia mensagens para esse tópico quando a atualização de uma shadow é rejeitada. Para obter mais informações, consulte /update/rejected (p. 669).

Tópico	Operações do cliente permitidas	Descrição
ShadowTopicPrefix/update/delta	Inscrever-se	O serviço Device Shadow envia mensagens para esse tópico quando uma diferença é detectada entre as seções desejada e relatada de uma shadow. Para obter mais informações, consulte /update/delta (p. 667).
ShadowTopicPrefix/update/documents	Inscrever-se	A AWS IoT publica um documento do estado nesse tópico sempre que a atualização de um shadow é feita com êxito. Para obter mais informações, consulte /update/documents (p. 668).

Tópicos de entrega de arquivos baseados em MQTT

Essas mensagens oferecem suporte a buffers de resposta no formato CBOR (Concise Binary Object Representation) e JavaScript JSON (Object Notation), dependendo da*formato da carga*do tópico.

formato da carga	Tipo de dados do formato de resposta
cbor	Concise Binary Object Representation (CBOR)
json	JavaScript Object Notation (JSON)

Tópico	Operações do cliente permitidas	Descrição
$aws/things/*ThingName*/streams/*StreamId*/dados/*formato da carga*	Inscrever-se	AWSA entrega de arquivos baseada em MQTT será publicada neste tópico se a solicitação “GetStream” de um dispositivo for aceita. A carga útil contém os dados do fluxo. Para obter mais informações, consulte O uso doAWS IoTEntrega de arquivos baseada em MQTT em dispositivos (p. 874).
$aws/things/*ThingName*/streams/*StreamId*/get/*formato da carga*	Publicar	Um dispositivo faz publicações nesse tópico para executar uma solicitação “GetStream”. Para obter mais informações, consulte O uso doAWS IoTEntrega de arquivos baseada em MQTT em dispositivos (p. 874).
$aws/things/*ThingName*/streams/*StreamId*/	Inscrever-se	AWSA entrega de arquivos baseada em MQTT será

Tópico	Operações do cliente permitidas	Descrição
descrição/*formato da carga*		publicada neste tópico se a solicitação “DescribeStream” de um dispositivo for aceita. A carga útil contém a descrição do fluxo. Para obter mais informações, consulte O uso doAWS IoTEntrega de arquivos baseada em MQTT em dispositivos (p. 874).
$aws/things/*ThingName*/streams/*StreamId*/describe/*formato da carga*	Publicar	Um dispositivo faz publicações nesse tópico para executar uma solicitação “DescribeStream”. Para obter mais informações, consulte O uso doAWS IoTEntrega de arquivos baseada em MQTT em dispositivos (p. 874).
$aws/things/*ThingName*/streams/*StreamId*/rejected/*formato da carga*	Inscrever-se	AWSA entrega de arquivos baseada em MQTT será publicada neste tópico se uma solicitação “DescribeStream” ou “GetStream” de um dispositivo for rejeitada. Para obter mais informações, consulte O uso doAWS IoTEntrega de arquivos baseada em MQTT em dispositivos (p. 874).

ARN do tópico reservado

Todos os ARNs de tópicos reservados (Amazon Resource Names) têm o seguinte formulário:

```
arn:aws:iot:aws-region:AWS-account-ID:topic/Topic
```

Por exemplo,`arn:aws:iot:us-west-2:123EXAMPLE456:topic/$aws/things/thingName/jobs/get/acceptedé` um ARN para o tópico reservado`$aws/things/thingName/jobs/get/accepted`.

endpoints configuráveis

> Note
>
> Esse recurso não está disponível no GovCloudRegiões da AWS.

comAWS IoT Core, você pode configurar e gerenciar os comportamentos de seus endpoints de dados usando configurações de domínio. Você pode gerar váriosAWS IoT Coreendpoints de dados e também personalize endpoints de dados com seus próprios nomes de domínio totalmente qualificados (e certificados de servidor associados) e autorizadores. Para obter mais informações sobre autorizadores personalizados, consultethe section called “Autenticação personalizada” (p. 319).

O AWS IoT Core usa a extensão TLS do Server Name Indication (SNI – Indicação de Nome de Servidor) para aplicar configurações de domínio. Os dispositivos devem usar essa extensão quando se conectarem. Eles também devem transmitir um nome de servidor idêntico ao nome de domínio especificado na

configuração de domínio.* Para testar esse serviço, use a versão v2 doAWS IoTSDKs de dispositivo dano GitHub

> **Note**
>
> Se você criar vários endpoints de dados em seuConta da AWS, eles compartilharãoAWS IoT Corerecursos como tópicos do MQTT, sombras de dispositivos e regras.

É possível usar configurações de domínio para simplificar tarefas como a seguinte.

- Migrar dispositivos para o AWS IoT
- Suporte a frotas de dispositivos heterogêneos mantendo configurações de domínio separadas para tipos de dispositivos separados.
- Manter a identidade da marca (por exemplo, pelo nome de domínio) ao migrar a infraestrutura do aplicativo ao AWS IoT.

É possível configurar um nome de domínio totalmente qualificado (FQDN), além do certificado de servidor associado. Você também pode associar um autorizador personalizado. Para obter mais informações, consulte Autenticação personalizada (p. 319).

> **Note**
>
> O AWS IoT usa a extensão TLS do Server Name Indication (SNI – Indicação de Nome de Servidor) para aplicar configurações de domínio. Os dispositivos devem usar essa extensão ao conectar e transmitir um nome de servidor idêntico ao nome de domínio especificado na configuração de domínio. Para testar esse serviço, use a versão v2 de cada SDK de dispositivo da AWS IoT no GitHub.

> **Note**
>
> Quando você fornece os certificados do servidor paraAWS IoTconfiguração de domínio personalizado, os certificados têm no máximo quatro nomes de domínio. Para obter mais informações, consulte Endpoints e cotas do AWS IoT Core.

Tópicos

- Criar e configurar domínios gerenciados pela AWS (p. 118)
- Criar e configurar domínios personalizados (p. 119)
- Gerenciar configurações de domínio (p. 122)

Criar e configurar domínios gerenciados pela AWS

Crie um endpoint configurável em um domínio gerenciado pela AWS usando a API CreateDomainConfiguration. Uma configuração de um domínio gerenciado pela AWS consiste no seguinte:

- `domainConfigurationName`— um nome definido pelo usuário que identifica a configuração de domínio.

  > **Note**
  >
  > Os nomes de configuração de domínio que começam com `IoT:` são reservados para endpoints padrão e não podem ser usados. Além disso, esse valor deve ser exclusivo da suaRegião da AWS.
- `defaultAuthorizerName`(opcional) — O nome do autorizador personalizado que deve ser usado no endpoint.
- `allowAuthorizerOverride`— um valor booliano que especifica se os dispositivos podem substituir o autorizador padrão especificando outro autorizador no cabeçalho HTTP da solicitação. Esse valor será exigido se um valor para `defaultAuthorizerName` for especificado.

- `serviceType`–AWS IoTNo momento, oferece suporte apenas a`DATA`Tipo de serviço. Quando você especifica`DATA`,AWS IoTO retorna um endpoint com um tipo de endpoint de`iot:Data-ATS`. Não é possível criar um endpoint `iot:Data` (VeriSign) configurável.

O comando da AWS CLI a seguir cria a configuração de domínio para um endpoint `Data`.

```
aws iot create-domain-configuration --domain-configuration-name "myDomainConfigurationName"
 --service-type "DATA"
```

Criar e configurar domínios personalizados

As configurações de domínio permitem especificar um nome de domínio totalmente qualificado (FQDN) para conectar-se ao AWS IoT. Os domínios personalizados permitem gerenciar seus próprios certificados de servidor para que seja possível gerenciar detalhes, como a autoridade de certificação (CA) raiz usada para assinar o certificado, o algoritmo de assinatura, a profundidade da cadeia de certificados e o ciclo de vida do certificado.

O fluxo de trabalho para definir uma configuração de domínio com um domínio personalizado consiste nos três estágios a seguir.

1. Registrar certificados de servidor no AWS Certificate Manager (p. 119)
2. Criar uma configuração de domínio (p. 120)
3. Criar registros DNS (p. 121)

Registrar certificados de servidor noAWSgerenciador de certificado

Antes de criar uma configuração de domínio com um domínio personalizado, é necessário registrar a cadeia de certificados de servidor noAWS Certificate Manager(ACM). É possível usar três tipos de certificados de servidor.

- Certificados públicos gerados pelo ACM (p. 120)
- Certificados externos assinados por uma CA pública (p. 120)
- Certificados externos assinados por uma CA privada (p. 120)

> **Note**
>
> O AWS IoT considera um certificado como assinado por uma CA pública se estiver incluído em pacote de CAs confiáveis do Mozilla.

Requisitos de certificado

ConsultePré-requisitos para importação de certificadospara os requisitos de importação de certificados para o ACM. Além desses requisitos,AWS IoT Coreadiciona os seguintes requisitos.

- O certificado de folha deve incluir conter aUso da chave estendidaextensão x509 v3 com um valor deServerAuth(Autenticação do servidor Web TLS). Se você solicitar o certificado do ACM, essa extensão será adicionada automaticamente.
- A profundidade máxima da cadeia de certificados é de 5 certificados.
- O tamanho máximo da cadeia de certificados é de 16 KB.

Usando um certificado para vários domínios

Se você planeja usar um certificado para abranger vários subdomínios, use um domínio curinga no campo common name (CN) ou Subject Alternative Names (SAN). Por exemplo, use **`*.iot.example.com`**

para abranger dev.iot.example.com, qa.iot.example.com e prod.iot.example.com. Cada FQDN exige sua própria configuração de domínio, mas mais de uma configuração de domínio pode usar o mesmo valor curinga. O CN ou o SAN devem abranger o FQDN que você deseja usar como domínio personalizado. Se as SANs estiverem presentes, a CN será ignorada e uma SAN deverá abranger o FQDN que você deseja usar como domínio personalizado. Essa abrangência pode ser uma correspondência exata ou uma correspondência curinga.

As seções a seguir descrevem como obter cada tipo de certificado. Cada recurso de certificado exige um Amazon Resource Name (ARN) registrado com o ACM usado ao criar a configuração de domínio.

Certificados públicos gerados pelo ACM

É possível gerar um certificado público para o domínio personalizado usando a API RequestCertificate. Ao gerar um certificado dessa maneira, o ACM valida a sua propriedade do domínio personalizado. Para obter mais informações, consulte Solicitar um certificado público no Guia do usuário do AWS Certificate Manager.

Certificados externos assinados por uma CA pública

Se você já tiver um certificado de servidor assinado por uma CA pública (uma CA incluída no pacote de CAs confiáveis do Mozilla), poderá importar a cadeia de certificados diretamente no ACM usando aImportCertificateAPI. Para saber mais sobre essa tarefa, sobre os pré-requisitos e sobre as exigências de formato do certificado, consulte Importar certificados.

Certificados externos assinados por uma CA privada

Se você já tiver um certificado de servidor assinado por uma CA privada ou autoassinado, será possível usar o certificado para criar a configuração de domínio, mas também será necessário criar um certificado público adicional no ACM para validar a propriedade do seu domínio. Para fazer isso, registre a cadeia de certificados de servidor no ACM usando aImportCertificateAPI. Para saber mais sobre essa tarefa, sobre os pré-requisitos e sobre as exigências de formato do certificado, consulte Importar certificados.

Após importar o certificado para o ACM, gere um certificado público para o domínio personalizado usando aRequestCertificateAPI. Ao gerar um certificado dessa maneira, o ACM valida a sua propriedade do domínio personalizado. Para obter mais informações, consulte Solicitar um certificado público. Ao criar a configuração de domínio, utilize esse certificado público como certificado de validação.

Criando uma configuração de domínio

Crie um endpoint configurável em um domínio personalizado usando aCreateDomainConfigurationAPI. Uma configuração de domínio para um domínio personalizado consiste no seguinte:

- `domainConfigurationName`— um nome definido pelo usuário que identifica a configuração de domínio.

  > **Note**
  >
  > Os nomes da configuração de domínio que começam com `IoT:` são reservados para endpoints padrão e não podem ser usados. Além disso, esse valor deve ser exclusivo da suaRegião da AWS.

- `domainName`— O FQDN que os dispositivos usam para conectar aoAWS IoT.

  > **Note**
  >
  > O AWS IoT utiliza a extensão TLS do Server Name Indication (SNI – Indicação de Nome de Servidor) para aplicar configurações de domínio. Os dispositivos devem usar essa extensão ao conectar e transmitir um nome de servidor idêntico ao nome de domínio especificado na configuração de domínio.

- `serverCertificateArns`— o ARN da cadeia de certificados de servidor registrada no ACM.AWS IoT CoreNo momento, oferece suporte a somente um certificado de servidor.

- `validationCertificateArn`— o ARN do certificado público gerado no ACM para validar a propriedade do domínio personalizado. Esse argumento não será necessário se você usar um certificado de servidor gerado pelo ACM ou assinado publicamente.
- `defaultAuthorizerName`(opcional) — O nome do autorizador personalizado que deve ser usado no endpoint.
- `allowAuthorizerOverride`— um valor booliano que especifica se os dispositivos podem substituir o autorizador padrão especificando outro autorizador no cabeçalho HTTP da solicitação. Esse valor será exigido se um valor para `defaultAuthorizerName` for especificado.
- `serviceType`–AWS IoTNo momento, oferece suporte apenas a`DATA`Tipo de serviço. Quando você especifica`DATA`,AWS IoTO retorna um endpoint com um tipo de endpoint de`iot:Data-ATS`.

O comando de AWS CLI a seguir cria uma configuração de domínio para iot.example.com.

```
aws iot create-domain-configuration --domain-configuration-name "myDomainConfigurationName"
 --service-type "DATA"
--domain-name "iot.example.com" --server-certificate-arns serverCertARN --validation-
certificate-arn validationCertArn
```

> **Note**
>
> Após criar a configuração de domínio, poderá levar até 60 minutos paraAWS IoTserve seus certificados de servidor personalizados.

Criar registros DNS

Depois de registrar sua cadeia de certificados e criar sua configuração de domínio, crie um registro de DNS para que o domínio personalizado aponte para um domínio da AWS IoT. Esse registro deve apontar para um endpoint do AWS IoT do tipo `iot:Data-ATS`. É possível obter o endpoint usando aDescribeEndpointAPI.

Os seguintes exemplos deAWS CLIO comando mostra como obter o endpoint.

```
aws iot describe-endpoint --endpoint-type iot:Data-ATS
```

Depois de obter o endpoint `iot:Data-ATS`, crie um registro `CNAME` do domínio personalizado para esse endpoint do AWS IoT. Se você criar vários domínios personalizados no mesmoConta da AWS, apelidá-los para este mesmo`iot:Data-ATS`.

Solução de problemas

Se você tiver problemas para conectar dispositivos a um domínio personalizado, certifique-se de queAWS IoT Coreaceitou e aplicou seu certificado de servidor. Você pode verificar seAWS IoT Coreaceitou seu certificado usando oAWS IoT Coreo console doAWS CLI.

Para usar aAWS IoT Core, navegue até oConfiguraçõese selecione o nome de configuração de domínio. NoDetalhes do certificado de servidor, verifique os detalhes de status e status. Se o certificado for inválido, substitua-o no ACM por um certificado que atenda aorequisitos de certificado (p. 119)Listado na seção anterior. Se o certificado tiver o mesmo ARN,AWS IoT Coreserá buscá-lo e aplicá-lo automaticamente.

Para verificar o status do certificado usando aAWS CLI, ligue para oDescribeDomainConfigurationAPI e especifique o nome de configuração de domínio.

> **Note**
>
> Se seu certificado for inválido,AWS IoT Corecontinuará a servir o último certificado válido.

Você pode verificar qual certificado está sendo veiculado em seu endpoint usando o seguinte comando openssl.

```
openssl s_client -connect custom-domain-name:8883 -showcerts -servername
custom-domain-name
```

Gerenciar configurações de domínio

É possível gerenciar os ciclos de vida de configurações existentes usando as APIs a seguir.

- ListDomainConfigurations
- DescribeDomainConfiguration
- UpdateDomainConfiguration
- DeleteDomainConfiguration

Visualizar configurações de domínio

Usar aListDomainConfigurationsAPI para retornar uma lista paginada de todas as configurações de domínio noConta da AWS. É possível ver os detalhes de uma configuração de domínio específica usando a API DescribeDomainConfiguration. Essa API usa um único parâmetro `domainConfigurationName` e retorna os detalhes da configuração especificada.

Atualizar configurações de domínio

Para atualizar o status do autorizador personalizado da configuração de domínio, use a API UpdateDomainConfiguration. É possível definir o status como `ENABLED` ou `DISABLED`. Se você desativar a configuração de domínio, os dispositivos conectados a esse domínio receberão um erro de autenticação.

> **Note**
>
> No momento, não é possível atualizar o certificado de servidor da configuração de domínio. Para alterar o certificado de uma configuração de domínio, é necessário excluí-la e recriá-la.

Exclusão de configurações de domínio

Antes de excluir uma configuração de domínio, use a API UpdateDomainConfiguration para definir o status como `DISABLED`. Isso ajuda a evitar a exclusão acidental do endpoint. Após desativar a configuração de domínio, exclua-a usando a API DeleteDomainConfiguration.

> **Note**
>
> Você deve colocarAWSdomínios gerenciados no`DISABLED`status por 7 dias antes de poder excluí-los. Você pode colocar domínios personalizados em`DISABLED`status e, em seguida, exclua-os imediatamente.

Após excluir uma configuração de domínio, a AWS IoT Core não fornecerá mais o certificado de servidor associado a esse domínio personalizado.

Rotação de certificados em domínios personalizados

Talvez seja necessário substituir periodicamente o certificado do servidor por um certificado atualizado. A taxa na qual você faz isso depende do período de validade do certificado. Se você gerou seu certificado de servidor usandoAWS Certificate Manager(ACM), você pode definir o certificado para renovar automaticamente. Quando o ACM renova seu certificado,AWS IoT CoreO seleciona automaticamente o novo certificado. Você não precisa realizar nenhuma ação adicional. Se você importou o certificado do servidor de uma fonte diferente, poderá girá-lo reimportando-o para o ACM. Para obter informações sobre como reimportar certificados, consulteReimportar um certificado.

> **Note**
>
> AWS IoT CoreSomente pega atualizações de certificado nas seguintes condições.

- O novo certificado tem o mesmo ARN do antigo.
- O novo certificado tem o mesmo algoritmo de assinatura, nome comum ou nome alternativo do assunto que o antigo.

Conexão aoAWS IoTEndpoints do FIPS

AWS IoTfornece endpoints que suportam oPadrão Federal de Processamento de Informações (FIPS) 140-2. Os endpoints compatíveis com FIPS são diferentes dos padrõesAWSEndpoints do . Para interagir com oAWS IoTde forma compatível com FIPS, você deve usar os endpoints descritos abaixo com seu cliente compatível com FIPS. OAWS IoTo console não é compatível com FIPS.

As seções a seguir descrevem como acessar o compatível com FIPSAWS IoTendpoints usando a API REST, um SDK da ou a.AWS CLI.

AWS IoT Core- endpoints do plano de controle

Compatível com os FIPSAWS IoT Core- plano de controleendpoints que suportam oAWS IoToperações e seus relacionadosComandos da CLI doestão listados emEndpoints do FIPS por serviço. DentroEndpoints do FIPS por serviço, encontre oAWS IoT Core- plano de controleserviço, e procure o endpoint para oRegião da AWS.

Para usar o endpoint compatível com FIPS quando você acessar oAWS IoToperações, use oAWSSDK ou a REST API com o endpoint apropriado para oRegião da AWS.

Para usar o endpoint compatível com FIPS quando você executaaws iotComandos da CLI do, adicione o--endpointparâmetro com o endpoint apropriado para oRegião da AWSPara o comando.

AWS IoT Core- endpoints do plano de dados

Compatível com os FIPSAWS IoT Core- plano de dadosOs endpoints são listados emEndpoints do FIPS por serviço. DentroEndpoints do FIPS por serviço, encontre oAWS IoT Core- plano de dadosserviço, e procure o endpoint para oRegião da AWS.

Você pode usar o endpoint compatível com FIPS para seuRegião da AWScom um cliente compatível com FIPS usando oAWS IoTSDK do dispositivo e fornecendo o endpoint para a função de conexão do SDK no lugar do padrão da sua contaAWS IoT Core- plano de dados. A função de conexão é específica para oAWS IoTSDK de dispositivo da. Para obter um exemplo de uma função de conexão, consulte aFunção de conexão noAWS IoTSDK de dispositivo da para Python.

> Note
>
> AWS IoTAs não são compatíveis com.Conta da AWSEspecíficoAWS IoT Core- plano de dadosendpoints compatíveis com FIPS. Recursos de serviço que exigem umConta da AWSEndpoint específico noIndicação de nome de servidor (SNI) (p. 381)As não podem ser usadas. Compatível com FIPSAWS IoT Core- plano de dadosendpoints não podem suportarCertificados de registro de várias contas (p. 299),Domínios personalizados (p. 119), eAutorizadores personalizados (p. 319).

AWS IoT Device Management- endpoints de dados de trabalhos

Compatível com os FIPSAWS IoT Device Management- dados de trabalhosOs endpoints são listados emEndpoints do FIPS por serviço. DentroEndpoints do FIPS por serviço, encontre oAWS IoT Device Management- dados de trabalhosserviço, e procure o endpoint para oRegião da AWS.

Para usar o FIPS compatívelAWS IoT Device Management- dados de trabalhosendpoint quando você executaaws iot-jobs-dataComandos da CLI do, adicione o--endpointparâmetro com o endpoint apropriado para oRegião da AWSPara o comando. Você também pode usar a API REST com esse endpoint.

Você pode usar o endpoint compatível com FIPS para seuRegião da AWScom um cliente compatível com FIPS usando oAWS IoTSDK do dispositivo e fornecendo o endpoint para a função de conexão do SDK no lugar do padrão da sua contaAWS IoT Device Management- dados de trabalhos. A função de conexão é específica para oAWS IoTSDK de dispositivo da. Para obter um exemplo de uma função de conexão, consulte aFunção de conexão noAWS IoTSDK de dispositivo da para Python.

AWS IoT Device Management- Endpoints do Fleet Hub

Compatível com os FIPSAWS IoT Device Management- Fleet Hubendpoints para usar comFleet Hub forAWS IoTGerenciamento de dispositivos Comandos da CLI doestão listados emEndpoints do FIPS por serviço. DentroEndpoints do FIPS por serviço, encontre oAWS IoT Device Management- Fleet Hubserviço, e procure o endpoint para oRegião da AWS.

Para usar o FIPS compatívelAWS IoT Device Management- Fleet Hubendpoint quando você executaaws iotfleethubComandos da CLI do, adicione o--endpointparâmetro com o endpoint apropriado para oRegião da AWSPara o comando. Você também pode usar a API REST com esse endpoint.

AWS IoT Device Management- endpoints de encapsulamento seguro

Compatível com os FIPSAWS IoT Device Management- Encapsulamento seguro doEndpoints para oAWS IoTAPI de encapsulamento seguroe o correspondenteComandos da CLI doestão listados emEndpoints do FIPS por serviço. DentroEndpoints do FIPS por serviço, encontre oAWS IoT Device Management- Encapsulamento seguro doserviço, e procure o endpoint para oRegião da AWS.

Para usar o FIPS compatívelAWS IoT Device Management- Encapsulamento seguro doendpoint quando você executaaws iotsecuretunnelingComandos da CLI do, adicione o--endpointparâmetro com o endpoint apropriado para oRegião da AWSPara o comando. Você também pode usar a API REST com esse endpoint.

Tutoriais do AWS IoT

OAWS IoTos tutoriais são divididos em dois caminhos de aprendizado para suportar dois objetivos diferentes. Escolha o melhor caminho de aprendizado para seu objetivo.

- Você quer construir um proof-of-concept para testar ou demonstrar umAWS IoTIdeia da solução

  Para demonstrar tarefas e aplicativos comuns de IoT usando oAWS IoTDevice Client em seus dispositivos, siga athe section called “Criando demonstrações com oAWS IoTCliente de dispositivo” (p. 125)Um caminho de aprendizado. OAWS IoTO Device Client fornece software de dispositivo com o qual você pode aplicar seus próprios recursos de nuvem para demonstrar um end-to-end solução com desenvolvimento mínimo.

  Para obter informações sobreAWS IoTDevice Client, consulte oAWS IoTCliente de dispositivo.
- Você quer aprender a criar software de produção para implantar sua solução

  Para criar seu próprio software de solução que atenda aos seus requisitos específicos usando umAWS IoTSDK do dispositivo, siga othe section called “Construindo soluções com oAWS IoTSDKs de dispositivo da” (p. 176)Um caminho de aprendizado.

  Para obter informações sobre o disponívelAWS IoTConsulte os SDKs de dispositivo, consulte??? (p. 1278). Para obter informações sobreAWSSDKs, consulteFerramentas para construirAWS.

AWS IoTopções de caminho de aprendizado tutorial

- Criando demonstrações com oAWS IoTCliente de dispositivo (p. 125)
- Construindo soluções com oAWS IoTSDKs de dispositivo da (p. 176)

Criando demonstrações com oAWS IoTCliente de dispositivo

Os tutoriais neste caminho de aprendizado orientam você pelas etapas para desenvolver software de demonstração usando oAWS IoTCliente de dispositivo. OAWS IoTO Device Client fornece software que é executado em seu dispositivo IoT para testar e demonstrar aspectos de uma solução de IoT que se baseiaAWS IoT.

O objetivo desses tutoriais é facilitar a exploração e a experimentação para que você possa se sentir confiante de queAWS IoToferece suporte à sua solução antes de desenvolver o software do dispositivo.

O que você aprenderá nesses tutoriais:

- Como preparar um Raspberry Pi para uso como um dispositivo de IoT comAWS IoT
- Como demonstrarAWS IoTrecursos usando oAWS IoTDevice Client em seu dispositivo

Neste caminho de aprendizado, você instalará oAWS IoTDevice Client em seu próprio Raspberry Pi e crie oAWS IoTrecursos na nuvem para demonstrar ideias de soluções de IoT. Enquanto os tutoriais neste caminho de aprendizado demonstram recursos usando um Raspberry Pi, eles explicam as metas e procedimentos para ajudá-lo a adaptá-los a outros dispositivos.

Pré-requisitos para criar demonstrações com oAWS IoTCliente de dispositivo

Esta seção descreve o que você precisa ter antes de iniciar os tutoriais neste caminho de aprendizado.

Para concluir os tutoriais neste caminho de aprendizado, você precisará de:

- Uma Conta da AWS

  Você pode usar o existenteConta da AWS, se você tiver um, mas talvez seja necessário adicionar funções ou permissões adicionais para usar oAWS IoTrecursos que esses tutoriais usam.

  Se você precisar criar um novoConta da AWS, consultethe section called “Configurar seuConta da AWS” (p. 19).

- Um Raspberry Pi ou dispositivo IoT compatível

  Os tutoriais usam umRaspberry Pporque vem em diferentes fatores de forma, é onipresente e é um dispositivo de demonstração relativamente barato. Os tutoriais foram testados noRaspberry Pi 3 modelo B+, oRaspberry Pi 4 modelo Be em uma instância do Amazon EC2 executando o Ubuntu Server 20.04 LTS (HVM). Para usar aAWS CLIe execute os comandos, Recomendamos que você use a versão mais recente do Raspberry Pi OS (Raspberry Pi OS (64 bits)ou o OS Lite). Versões anteriores do sistema operacional podem funcionar, mas não o testamos.

  > Note
  >
  > Os tutoriais explicam os objetivos de cada etapa para ajudá-lo a adaptá-los ao hardware de IoT que não os experimentamos; no entanto, eles não descrevem especificamente como adaptá-los a outros dispositivos.

- Familiaridade com o sistema operacional do dispositivo IoT

  As etapas nesses tutoriais pressupõem que você esteja familiarizado com o uso de comandos e operações básicos do Linux a partir da interface de linha de comando suportada por um Raspberry Pi. Se você não estiver familiarizado com essas operações, talvez queira dar a si mesmo mais tempo para concluir os tutoriais.

  Para concluir esses tutoriais, você já deve entender como:

  - Execute com segurança operações básicas do dispositivo, como montagem e conexão de componentes, conexão do dispositivo às fontes de energia necessárias e instalação e remoção de cartões de memória.
  - Carregue e baixe o software e os arquivos do sistema para o dispositivo. Se o dispositivo não usar um dispositivo de armazenamento removível, como um cartão microSD, você precisará saber como se conectar ao seu dispositivo e carregar e baixar software e arquivos do sistema para o dispositivo.
  - Connect seu dispositivo às redes nas quais você planeja usá-lo.
  - Connect ao seu dispositivo a partir de outro computador usando um terminal SSH ou programa similar.
  - Use uma interface de linha de comando para criar, copiar, mover, renomear e definir as permissões de arquivos e diretórios no dispositivo.
  - Instale novos programas no dispositivo.
  - Transfira arquivos de e para o dispositivo usando ferramentas como FTP ou SCP.

- Um ambiente de desenvolvimento e teste para sua solução de IoT

  Os tutoriais descrevem o software e o hardware necessários; no entanto, os tutoriais assumem que você poderá executar operações que podem não ser descritas explicitamente. Exemplos desse hardware e operações incluem:

  - Um computador host local para baixar e armazenar arquivos

Para o Raspberry Pi, geralmente é um computador pessoal ou laptop que pode ler e gravar em cartões de memória microSD. O computador host local deve:

- Esteja conectado à Internet.
- Tenha aAWS CLIInstalado e configurado.
- Tenha um navegador da Web que suporte oAWSconsole do .

- Uma maneira de conectar seu computador host local ao dispositivo para se comunicar com ele, inserir comandos e transferir arquivos

  No Raspberry Pi, isso geralmente é feito usando SSH e SCP do computador host local.
- Um monitor e teclado para se conectar ao seu dispositivo IoT

  Eles podem ser úteis, mas não são necessários para concluir os tutoriais.
- Uma maneira de o computador host local e seus dispositivos IoT se conectarem à Internet

  Isso pode ser uma conexão de rede sem fio ou cabeada a um roteador ou gateway conectado à Internet. O host local também deve ser capaz de se conectar ao Raspberry Pi. Isso pode exigir que eles estejam na mesma rede local. Os tutoriais não podem mostrar como configurar isso para sua configuração específica de dispositivo ou dispositivo, mas mostram como você pode testar essa conectividade.
- Acesso ao roteador da rede local para visualizar os dispositivos conectados

  Para concluir os tutoriais neste caminho de aprendizado, você precisará encontrar o endereço IP do seu dispositivo IoT.

  Em uma rede local, isso pode ser feito acessando a interface de administração do roteador de rede ao qual seus dispositivos se conectam. Se você puder atribuir um endereço IP fixo ao seu dispositivo no roteador, poderá simplificar a reconexão após cada vez que o dispositivo for reiniciado.

  Se você tiver um teclado e um monitor conectados ao dispositivo,ifconfigpode exibir o endereço IP do dispositivo.

  Se nenhuma delas for uma opção, você precisará encontrar uma maneira de identificar o endereço IP do dispositivo após cada vez que ele for reiniciado.

Depois de ter todos os seus materiais, continuethe section called “Preparando seus dispositivos para oAWS IoTCliente do dispositivo” (p. 127).

Tutoriais neste caminho de aprendizado

- Tutorial: Preparando seus dispositivos para oAWS IoTCliente do dispositivo (p. 127)
- Tutorial: Instalar e configurar oAWS IoTCliente do dispositivo (p. 137)
- Tutorial: Demonstrar comunicação de mensagens MQTT com oAWS IoTCliente do dispositivo (p. 146)
- Tutorial: Demonstrar ações remotas (trabalhos) com oAWS IoTCliente do dispositivo (p. 160)
- Tutorial: Limpeza após a execução doAWS IoTTutoriais do dispositivo cliente (p. 169)

Tutorial: Preparando seus dispositivos para oAWS IoTCliente do dispositivo

Este tutorial orienta você pela inicialização do seu Raspberry Pi para prepará-lo para os tutoriais subsequentes neste caminho de aprendizado.

O objetivo deste tutorial é instalar a versão atual do sistema operacional do dispositivo e garantir que você possa se comunicar com seu dispositivo no contexto do ambiente de desenvolvimento.

Para iniciar este tutorial:

- Tenha os itens listados emthe section called “Pré-requisitos para criar demonstrações com oAWS IoTCliente de dispositivo” (p. 126)Disponível e pronto para uso.

A conclusão deste tutorial leva cerca de 90 minutos.

Quando terminar este tutorial:

- Seu dispositivo IoT terá um sistema operacional atualizado.
- Seu dispositivo IoT terá o software adicional necessário para os tutoriais subsequentes.
- Você saberá que seu dispositivo tem conectividade com a Internet.
- Você terá instalado um certificado necessário no dispositivo.

Depois de concluir este tutorial, o próximo tutorial prepara seu dispositivo para as demonstrações que usam oAWS IoTCliente do dispositivo.

Procedimentos neste tutorial

- Etapa 1: Instale e atualize o sistema operacional do dispositivo (p. 128)
- Etapa 2: Instale e verifique o software necessário em seu dispositivo (p. 131)
- Etapa 3: Teste seu dispositivo e salve o certificado da Amazon CA (p. 134)

Etapa 1: Instale e atualize o sistema operacional do dispositivo

Os procedimentos desta seção descrevem como inicializar o cartão microSD que o Raspberry Pi usa para sua unidade de sistema. O cartão microSD do Raspberry Pi contém seu software de sistema operacional (SO), bem como espaço para armazenamento de arquivos de aplicativos. Se você não estiver usando um Raspberry Pi, siga as instruções do dispositivo para instalar e atualizar o software do sistema operacional do dispositivo.

Depois de concluir esta seção, você poderá iniciar o dispositivo IoT e conectar-se a ele a partir do programa de terminal em seu computador host local.

Equipamento necessário:

- Seu ambiente local de desenvolvimento e teste
- Um Raspberry Pi que ou o seu dispositivo IoT, que pode se conectar à Internet
- Um cartão de memória microSD com pelo menos 8 GB de capacidade ou armazenamento suficiente para o sistema operacional e o software necessário.

  Note

  Ao selecionar um cartão microSD para esses exercícios, escolha um que seja o maior necessário, mas o menor possível.
  Um pequeno cartão SD será mais rápido para fazer backup e atualizar. No Raspberry Pi, você não precisará de mais de um cartão microSD de 8 GB para esses tutoriais. Se você precisar de mais espaço para seu aplicativo específico, os arquivos de imagem menores que você salvar nesses tutoriais poderão redimensionar o sistema de arquivos em um cartão maior para usar todo o espaço suportado do cartão escolhido.

Equipamento opcional:

- Um teclado USB conectado ao Raspberry Pi
- Um monitor HDMI e cabo para conectar o monitor ao Raspberry Pi

Procedimentos nesta seção:

- Carregar o sistema operacional do dispositivo no cartão microSD (p. 129)
- Inicie seu dispositivo IoT com o novo sistema operacional (p. 129)
- Connect seu computador host local ao dispositivo (p. 130)

Carregar o sistema operacional do dispositivo no cartão microSD

Este procedimento usa o computador host local para carregar o sistema operacional do dispositivo em um cartão microSD.

> Note
>
> Se o dispositivo não usar um meio de armazenamento removível para o sistema operacional, instale o sistema operacional usando o procedimento desse dispositivo e continue athe section called “Inicie seu dispositivo IoT com o novo sistema operacional” (p. 129).

Para instalar o sistema operacional em seu Raspberry Pi

1. Em seu computador host local, baixe e descompacte a imagem do sistema operacional Raspberry Pi que você deseja usar. As versões mais recentes estão disponíveis emhttps://www.raspberrypi.com/software/operating-systems/

   Escolhendo uma versão do Raspberry Pi OS

   Este tutorial usa oRaspberry Pi OS Liteversão porque é a versão mais pequena que suporta esses tutoriais neste caminho de aprendizado. Esta versão do Raspberry Pi OS tem apenas uma interface de linha de comando e não tem uma interface gráfica de usuário. Uma versão do sistema operacional Raspberry Pi mais recente com uma interface gráfica de usuário também funcionará com esses tutoriais; no entanto, os procedimentos descritos neste caminho de aprendizado usam apenas a interface de linha de comando para executar operações no Raspberry Pi.
2. Insira seu cartão microSD no computador host local.
3. Usando uma ferramenta de imagem de cartão SD, grave o arquivo de imagem do SO descompactado no cartão microSD.
4. Depois de escrever a imagem do Raspberry Pi OS no cartão microSD:

   a. Abra a partição BOOT no cartão microSD em uma janela de linha de comando ou janela do explorador de arquivos.
   b. Na partição BOOT do cartão microSD, no diretório raiz, crie um arquivo vazio chamado`ssh`sem extensão de arquivo e sem conteúdo. Isso diz ao Raspberry Pi para habilitar as comunicações SSH na primeira vez que ele começa.
5. Ejete o cartão microSD e remova-o com segurança do computador host local.

Seu cartão microSD está pronto parathe section called “Inicie seu dispositivo IoT com o novo sistema operacional” (p. 129).

Inicie seu dispositivo IoT com o novo sistema operacional

Este procedimento instala o cartão microSD e inicia seu Raspberry Pi pela primeira vez usando o sistema operacional baixado.

Para iniciar o dispositivo IoT com o novo sistema operacional

1. Com a energia desconectada do dispositivo, insira o cartão microSD da etapa anterior,the section called “Carregar o sistema operacional do dispositivo no cartão microSD” (p. 129), no Raspberry Pi.
2. Connect o dispositivo a uma rede com fio.
3. Esses tutoriais interagirão com seu Raspberry Pi do seu computador host local usando um terminal SSH.

   Se você também quiser interagir com o dispositivo diretamente, você pode:

   a. Connect um monitor HDMI a ele para assistir às mensagens do console do Raspberry Pi antes de poder conectar a janela do terminal no computador host local ao seu Raspberry Pi.
   b. Connect um teclado USB a ele se você quiser interagir diretamente com o Raspberry Pi.
4. Connect a energia ao Raspberry Pi e aguarde cerca de um minuto até que ele seja inicializado.

   Se você tiver um monitor conectado ao seu Raspberry Pi, poderá assistir ao processo de inicialização nele.
5. Descubra o endereço IP do seu dispositivo:

   - Se você conectou um monitor HDMI ao Raspberry Pi, o endereço IP aparecerá nas mensagens exibidas no monitor
   - Se você tiver acesso ao roteador ao qual o Raspberry Pi se conecta, você poderá ver o endereço dele na interface de administração do roteador.

Depois que tiver o endereço IP do Raspberry Pi, você estará pronto parathe section called “Connect seu computador host local ao dispositivo” (p. 130).

Connect seu computador host local ao dispositivo

Este procedimento usa o programa de terminal em seu computador host local para se conectar ao Raspberry Pi e alterar sua senha padrão.

Para conectar seu computador host local ao dispositivo

1. No computador host local, abra o programa de terminal SSH:

   - Windows: `PuTTY`
   - Linux/macOS:`Terminal`

   > **Note**
   >
   > O PuTTY não é instalado automaticamente no Windows. Se não estiver no computador, talvez seja necessário fazer download e instalá-lo.
2. Connect o programa de terminal ao endereço IP do Raspberry Pi e faça login usando suas credenciais padrão.

```
username: pi
password: raspberry
```

3. Depois de fazer login no seu Raspberry Pi, altere a senha do`pi`usuário.

```
passwd
```

   Siga as instruções para alterar a senha.

```
Changing password for pi.
Current password: raspberry
New password: YourNewPassword
Retype new password: YourNewPassword
passwd: password updated successfully
```

Depois de ter o prompt de linha de comando do Raspberry Pi na janela do terminal e alterar a senha, você estará pronto para continuarthe section called “Etapa 2: Instale e verifique o software necessário em seu dispositivo” (p. 131).

Etapa 2: Instale e verifique o software necessário em seu dispositivo

Os procedimentos nesta seção continuam dea seção anterior (p. 128)para atualizar o sistema operacional do Raspberry Pi e instalar o software no Raspberry Pi que será usado na próxima seção para criar e instalar oAWS IoTCliente do dispositivo.

Depois de concluir esta seção, seu Raspberry Pi terá um sistema operacional atualizado, o software exigido pelos tutoriais neste caminho de aprendizado, e ele será configurado para sua localização.

Equipamento necessário:

- Seu ambiente local de desenvolvimento e teste dea seção anterior (p. 128)
- O Raspberry Pi que você usou ema seção anterior (p. 128)
- O cartão de memória microSD daa seção anterior (p. 128)

> **Note**
>
> O Raspberry Pi Model 3+ e o Raspberry Pi Model 4 podem executar todos os comandos descritos neste caminho de aprendizado. Se o seu dispositivo IoT não puder compilar software ou executar oAWS Command Line Interface, talvez seja necessário instalar os compiladores necessários em seu computador host local para criar o software e, em seguida, transferi-lo para o dispositivo IoT. Para obter mais informações sobre como instalar e criar software para seu dispositivo, consulte a documentação do software do dispositivo.

Procedimentos nesta seção:

- Atualizar o software do sistema operacional (p. 131)
- Instale os aplicativos e bibliotecas necessários (p. 133)
- (Opcional) Salve a imagem do cartão microSD (p. 133)

Atualizar o software do sistema operacional

Este procedimento atualiza o software do sistema operacional.

Para atualizar o software do sistema operacional no Raspberry Pi

Execute estas etapas na janela do terminal do computador host local.

1. Insira estes comandos para atualizar o software do sistema no seu Raspberry Pi.

```
sudo apt-get -y update
```

```
sudo apt-get -y upgrade
sudo apt-get -y autoremove
```

2. Atualize as configurações de localidade e fuso horário do Raspberry Pi (opcional).

   Digite este comando para atualizar as configurações de localidade e fuso horário do dispositivo.

   ```
   sudo raspi-config
   ```

   a. Para definir a localidade do dispositivo:

      i. NoFerramenta de configuração de software Raspberry Pi (raspi-config)Selecione a opção5.

         `5 Localisation Options Configure language and regional settings`

         Usar aTabchave para mover para<Select>,Digite e pressione ospace bar.

      ii. No menu de opções de localização, escolha a opçãoL1.

         `L1 Locale Configure language and regional settings`

         Usar aTabchave para mover para<Select>,Digite e pressione ospace bar.

      iii. Na lista de opções de localidade, escolha os locais que você deseja instalar em seu Raspberry Pi usando as teclas de seta para rolar e ospace barpara marcar aqueles que você quer.

         Nos Estados Unidos,**`en_US.UTF-8`**é um bom para escolher.

      iv. Depois de selecionar as localidades do seu dispositivo, use oTabchave para escolher<OK>Digite e pressione ospace barPara exibir oConfigurar localidadesPágina de confirmação.

   b. Para definir o fuso horário do dispositivo:

      i. Noraspi-configSelecione a opção5.

         `5 Localisation Options Configure language and regional settings`

         Usar aTabchave para mover para<Select>,Digite e pressione ospace bar.

      ii. No menu de opções de localização, use a tecla de seta para escolher a opçãoL2:

         `L2 time zone Configure time zone`

         Usar aTabchave para mover para<Select>,Digite e pressione ospace bar.

      iii. NoConfigurando tzdatamenu, escolha sua área geográfica na lista.

         Usar aTabchave para mover para<OK>Digite e pressione ospace bar.

      iv. Na lista de cidades, use as teclas de seta para escolher uma cidade em seu fuso horário.

         Para definir o fuso horário, use oTabchave para mover para<OK>Digite e pressione ospace bar.

   c. Quando você terminar de atualizar as configurações, use oTabchave para mover para<Finish>Digite e pressione ospace barPara fechar oraspi-configaplicativo.

3. Insira este comando para reiniciar o Raspberry Pi.

   ```
   sudo shutdown -r 0
   ```

4. Aguarde a reinicialização do Raspberry Pi.
5. Depois que o Raspberry Pi for reiniciado, reconecte a janela do terminal no computador host local ao Raspberry Pi.

Seu software do sistema Raspberry Pi agora está configurado e você está pronto para continuarthe section called “Instale os aplicativos e bibliotecas necessários” (p. 133).

Instale os aplicativos e bibliotecas necessários

Este procedimento instala o software e as bibliotecas do aplicativo que os tutoriais subsequentes usam.

Se você estiver usando um Raspberry Pi ou se puder compilar o software necessário em seu dispositivo IoT, execute estas etapas na janela do terminal no computador host local. Se você precisar compilar software para o dispositivo IoT em seu computador host local, consulte a documentação do software do dispositivo IoT para obter informações sobre como executar essas etapas no dispositivo.

Para instalar o software do aplicativo e as bibliotecas no seu Raspberry Pi

1. Digite este comando para instalar o software e as bibliotecas do aplicativo.

```
sudo apt-get -y install build-essential libssl-dev cmake unzip git python3-pip
```

2. Insira esses comandos para confirmar se a versão correta do software foi instalada.

```
gcc --version
cmake --version
openssl version
git --version
```

3. Confirme se essas versões do software do aplicativo estão instaladas:

   - `gcc`: 9.3.0 ou posterior
   - `cmake`: 3.10.x ou posterior
   - `OpenSSL`: 1.1.1 ou posterior
   - `git`: 2.20.1 ou posterior

Se o Raspberry Pi tiver versões aceitáveis do software aplicativo necessário, você estará pronto para continuarthe section called “(Opcional) Salve a imagem do cartão microSD” (p. 133).

(Opcional) Salve a imagem do cartão microSD

Ao longo dos tutoriais neste caminho de aprendizado, você encontrará esses procedimentos para salvar uma cópia da imagem do cartão microSD do Raspberry Pi em um arquivo no computador host local. Embora incentivados, eles não são tarefas necessárias. Ao salvar a imagem do cartão microSD onde sugerido, você pode pular os procedimentos que precedem o ponto de salvamento neste caminho de aprendizado, o que pode economizar tempo se você encontrar a necessidade de repetir alguma coisa. A consequência de não salvar a imagem do cartão microSD periodicamente é que talvez seja necessário reiniciar os tutoriais no caminho de aprendizado desde o início se o cartão microSD estiver danificado ou se você acidentalmente configurar um aplicativo ou suas configurações incorretamente.

Neste ponto, o cartão microSD do Raspberry Pi possui um sistema operacional atualizado e o software básico do aplicativo carregado. Você pode economizar o tempo necessário para concluir as etapas anteriores, salvando o conteúdo do cartão microSD em um arquivo agora. Ter a imagem atual da imagem do cartão microSD do seu dispositivo permite que você comece a partir deste ponto para continuar ou repetir um tutorial ou procedimento sem a necessidade de instalar e atualizar o software do zero.

Para salvar a imagem do cartão microSD em um arquivo

1. Digite este comando para desligar o Raspberry Pi.

```
sudo shutdown -h 0
```

2. Depois que o Raspberry Pi desligar completamente, remova sua energia.
3. Remova o cartão microSD do Raspberry Pi.
4. No computador host local:
   a. Insira o cartão microSD.
   b. Usando a ferramenta de imagem do cartão SD, salve a imagem do cartão microSD em um arquivo.
   c. Depois que a imagem do cartão microSD for salva, ejete o cartão do computador host local.
5. Com a energia desconectada do Raspberry Pi, insira o cartão microSD no Raspberry Pi.
6. Aplique energia ao Raspberry Pi.
7. Depois de esperar cerca de um minuto, no computador host local, reconecte a janela do terminal no computador host local que estava conectado ao Raspberry Pi. e, em seguida, faça login no Raspberry Pi.

Etapa 3: Teste seu dispositivo e salve o certificado da Amazon CA

Os procedimentos nesta seção continuam dea seção anterior (p. 131)para instalar oAWS Command Line Interfacee o certificado da Autoridade de Certificação usado para autenticar suas conexões comAWS IoT Core.

Depois de concluir esta seção, você saberá que seu Raspberry Pi possui o software de sistema necessário para instalar oAWS IoTCliente de dispositivo e que tem uma conexão funcional com a Internet.

Equipamento necessário:

- Seu ambiente local de desenvolvimento e teste dea seção anterior (p. 131)
- O Raspberry Pi que você usou ema seção anterior (p. 131)
- O cartão de memória microSD daa seção anterior (p. 131)

Procedimentos nesta seção:

- Instalar a AWS Command Line Interface (p. 134)
- Configurar as credenciais da Conta da AWS (p. 135)
- Faça download do certificado CA raiz da Amazon (p. 136)
- (Opcional) Salve a imagem do cartão microSD (p. 136)

Instalar a AWS Command Line Interface

Este procedimento instala oAWS CLINo Raspberry Pi.

Se você estiver usando um Raspberry Pi ou se puder compilar software em seu dispositivo IoT, execute estas etapas na janela do terminal no computador host local. Se você precisar compilar software para o dispositivo IoT em seu computador host local, revise a documentação do software do dispositivo IoT para obter informações sobre as bibliotecas necessárias.

Para instalar oAWS CLIno Raspberry Pi

1. Execute esses comandos para fazer download doAWS CLI.

```
export PATH=$PATH:~/.local/bin # configures the path to include the directory with the
 AWS CLI
```

```
git clone https://github.com/aws/aws-cli.git # download the AWS CLI code from GitHub
cd aws-cli && git checkout v2 # go to the directory with the repo and checkout version
 2
pip3 install -r requirements.txt # install the prerequisite software
```

2. Execute este comando para instalar oAWS CLI. Esse comando pode levar até 15 minutos para ser concluído.

```
pip3 install . # install the AWS CLI
```

3. Execute este comando para confirmar se a versão correta doAWS CLIfoi instalado.

```
aws --version
```

   A versão doAWS CLIDeve ser 2.2 ou posterior.

Se oAWS CLIExibiu sua versão atual, você está pronto para continuar athe section called “Configurar as credenciais da Conta da AWS” (p. 135).

Configurar as credenciais da Conta da AWS

Neste procedimento, você obteráConta da AWScredenciais e adicione-as para uso em seu Raspberry Pi.

Para adicionar seuConta da AWScredenciais para o dispositivo

1. Obter umAccess Key IDeSecret Access Key (Chave de acesso secreta)do seuConta da AWSpara autenticar oAWS CLIno dispositivo.

   Se você é novo noAWSIAM,https://aws.amazon.com/premiumsupport/knowledge-center/create-access-key/descreve o processo a ser executado naAWSconsole para criarAWSCredenciais do IAM para usar em seu dispositivo.
2. Na janela do terminal do computador host local que está conectado ao seu Raspberry Pi. e com oAccess Key IDeSecret Access Key (Chave de acesso secreta)credenciais para seu dispositivo:

   a. Execute aAWSConfigurar aplicativo com este comando:

   ```
   aws configure
   ```

   b. Insira suas credenciais e informações de configuração quando solicitado:

   ```
   AWS Access Key ID: your Access Key ID
   AWS Secret Access Key: your Secret Access Key
   Default region name: your Região da AWS code
   Default output format: json
   ```
3. Execute este comando para testar o acesso do dispositivo aoConta da AWSeAWS IoT Core.

```
aws iot describe-endpoint --endpoint-type iot:Data-ATS
```

   Ele deve devolver seuConta da AWSespecífico doAWS IoTendpoint de dados, como este exemplo:

```
{
    "endpointAddress": "a3EXAMPLEffp-ats.iot.us-west-2.amazonaws.com"
}
```

Se você vir seuConta da AWSespecífico doAWS IoTendpoint de dados, seu Raspberry Pi tem a conectividade e as permissões para continuarthe section called “Faça download do certificado CA raiz da Amazon” (p. 136).

> **Important**
>
> SuasConta da AWScredenciais agora são armazenadas no cartão microSD em seu Raspberry Pi. Embora isso faça interações futuras comAWSfácil para você e o software que você criará nesses tutoriais, eles também serão salvos e duplicados em qualquer imagem de cartão microSD que você fizer após esta etapa por padrão.
> Para proteger a segurança do seuConta da AWScredenciais, antes de salvar mais imagens de cartão microSD, considere apagar as credenciais executando`aws configure`novamente e inserindo caracteres aleatórios para oAccess Key IDeSecret Access Key (Chave de acesso secreta)para evitar que seuConta da AWScredenciais de comprometidas.
> Se você achar que salvou seuConta da AWScredenciais inadvertidamente, você pode desativá-las noAWSConsole do IAM.

Faça download do certificado CA raiz da Amazon

Este procedimento baixa e salva uma cópia de um certificado da Autoridade de certificação raiz da Amazon (CA). O download deste certificado o salva para uso nos tutoriais subsequentes e também testa a conectividade do seu dispositivo comAWSServiços da .

Para fazer download do certificado CA raiz da Amazon

1. Execute este comando para criar um diretório para o certificado.

```
mkdir ~/certs
```

2. Execute este comando para fazer download do certificado CA raiz da Amazon.

```
curl -o ~/certs/AmazonRootCA1.pem https://www.amazontrust.com/repository/
AmazonRootCA1.pem
```

3. Execute esses comandos para definir o acesso ao diretório de certificados e ao arquivo.

```
chmod 745 ~
chmod 700 ~/certs
chmod 644 ~/certs/AmazonRootCA1.pem
```

4. Execute este comando para ver o arquivo de certificado da CA no novo diretório.

```
ls -l ~/certs
```

   Você deve ver uma entrada como esta. A data e a hora serão diferentes; no entanto, o tamanho do arquivo e todas as outras informações devem ser iguais às mostradas aqui.

```
-rw-r--r-- 1 pi pi 1188 Oct 28 13:02 AmazonRootCA1.pem
```

   Se o tamanho do arquivo não for`1188`, verifique ocurlparâmetros de comando. Você pode ter baixado um arquivo incorreto.

(Opcional) Salve a imagem do cartão microSD

Neste ponto, o cartão microSD do Raspberry Pi possui um sistema operacional atualizado e o software básico do aplicativo carregado.

Para salvar a imagem do cartão microSD em um arquivo

1. Na janela do terminal do computador host local, limpe oAWSCredenciais da .

   a. Execute aAWSConfigurar aplicativo com este comando:

   ```
   aws configure
   ```

   b. Substitua suas credenciais quando solicitado. Você pode sairNome da região padrãoeFormato de saída padrãocomo eles são pressionandoDigite.

   ```
   AWS Access Key ID [****************YT2H]: XYXYXYXYX
   AWS Secret Access Key [****************9plH]: XYXYXYXYX
   Default region name [us-west-2]:
   Default output format [json]:
   ```

2. Digite este comando para desligar o Raspberry Pi.

   ```
   sudo shutdown -h 0
   ```

3. Depois que o Raspberry Pi desligar completamente, remova o conector de alimentação.
4. Remova o cartão microSD do seu dispositivo.
5. No computador host local:

   a. Insira o cartão microSD.
   b. Usando a ferramenta de imagem do cartão SD, salve a imagem do cartão microSD em um arquivo.
   c. Depois que a imagem do cartão microSD for salva, ejete o cartão do computador host local.
6. Com a energia desconectada do Raspberry Pi, insira o cartão microSD no Raspberry Pi.
7. Aplique energia ao dispositivo.
8. Após cerca de um minuto, no computador host local, reinicie a sessão da janela do terminal e faça login no dispositivo.

   Não insira novamente seuConta da AWScredenciais ainda.

Depois de reiniciar e fazer login no seu Raspberry Pi, você estará pronto para continuarthe section called “Instalar e configurar oAWS IoTCliente do dispositivo” (p. 137).

Tutorial: Instalar e configurar oAWS IoTCliente do dispositivo

Este tutorial orienta você pela instalação e configuração doAWS IoTDevice Client e a criação deAWS IoTrecursos que você usará nesta e em outras demonstrações.

Para iniciar este tutorial:

- Tenha seu computador host local e o Raspberry Pi daO tutorial anterior (p. 127)Pronto.

Este tutorial pode levar até 90 minutos para ser concluído.

Quando terminar este tópico:

- Seu dispositivo IoT estará pronto para uso em outrosAWS IoTDemonstrações do Device
- Você provisionará seu dispositivo IoT emAWS IoT Core.

- Você terá baixado e instalado oAWS IoTDevice Client em seu dispositivo.
- Você salvará uma imagem do cartão microSD do seu dispositivo que pode ser usada em tutoriais subsequentes.

Equipamento necessário:

- Seu ambiente local de desenvolvimento e teste deA seção anterior (p. 134)
- O Raspberry Pi que você usou emA seção anterior (p. 134)
- O cartão de memória microSD do Raspberry Pi que você usouA seção anterior (p. 134)

Procedimentos neste tutorial

- Etapa 1: Faça download e salve oAWS IoTCliente do dispositivo (p. 138)
- (Opcional) Salve a imagem do cartão microSD (p. 139)
- Etapa 2: Provisão do Raspberry Pi emAWS IoT (p. 140)
- Etapa 3: Configurar aAWS IoTCliente de dispositivo para testar a conectividade (p. 144)

Etapa 1: Faça download e salve oAWS IoTCliente do dispositivo

Os procedimentos nesta seção baixam oAWS IoTDevice Client, compile-o e instale-o em seu Raspberry Pi. Depois de testar a instalação, você pode salvar a imagem do cartão microSD do Raspberry Pi para usar mais tarde quando quiser tentar os tutoriais novamente.

Procedimentos nesta seção:

- Faça download e crie oAWS IoTCliente do dispositivo (p. 138)
- Crie os diretórios usados pelos tutoriais (p. 139)

Faça download e crie oAWS IoTCliente do dispositivo

Este procedimento instala oAWS IoTCliente de dispositivo no Raspberry Pi.

Execute esses comandos na janela do terminal em seu computador host local conectado ao seu Raspberry Pi.

Para instalar oAWS IoTCliente de dispositivo no Raspberry Pi

1. Insira esses comandos para fazer download e criar oAWS IoTCliente de dispositivo no Raspberry Pi.

```
cd ~
git clone https://github.com/awslabs/aws-iot-device-client aws-iot-device-client
mkdir ~/aws-iot-device-client/build && cd ~/aws-iot-device-client/build
cmake ../
```

2. Execute este comando para criar oAWS IoTCliente do dispositivo. Esse comando pode levar até 15 minutos para ser concluído.

```
cmake --build . --target aws-iot-device-client
```

   As mensagens de aviso exibidas como oAWS IoTAs compilações do Device Client podem ser ignoradas.

   Esses tutoriais foram testados com oAWS IoTDevice Client construído emgcc, versão (Raspbian 10.2.1-6+rpi1) 10.2.1 20210110 na versão de 30 de outubro de 2021 do Raspberry Pi OS (bullseye)

emgcc, versão (Raspbian 8.3.0-6+rpi1) 8.3.0 na versão de 7 de maio de 2021 do Raspberry Pi OS (buster).

3. Depois doAWS IoTO Device Client termina de construir, testá-lo executando este comando.

```
./aws-iot-device-client --help
```

Se vir a ajuda da linha de comando doAWS IoTDevice Client, oAWS IoTO Device Client foi criado com sucesso e está pronto para você usar.

Crie os diretórios usados pelos tutoriais

Este procedimento cria os diretórios no Raspberry Pi que serão usados para armazenar os arquivos usados pelos tutoriais neste caminho de aprendizado.

Para criar os diretórios usados pelos tutoriais neste caminho de aprendizado:

1. Execute esses comandos para criar os diretórios necessários.

```
mkdir ~/dc-configs
mkdir ~/policies
mkdir ~/messages
mkdir ~/certs/testconn
mkdir ~/certs/pubsub
mkdir ~/certs/jobs
```

2. Execute esses comandos para definir as permissões nos novos diretórios.

```
chmod 745 ~
chmod 700 ~/certs/testconn
chmod 700 ~/certs/pubsub
chmod 700 ~/certs/jobs
```

Depois de criar esses diretórios e definir a permissão, continue athe section called “(Opcional) Salve a imagem do cartão microSD” (p. 139).

(Opcional) Salve a imagem do cartão microSD

Neste ponto, o cartão microSD do Raspberry Pi possui um sistema operacional atualizado, o software básico do aplicativo e oAWS IoTCliente do dispositivo.

Se você quiser voltar para experimentar esses exercícios e tutoriais novamente, você pode pular os procedimentos anteriores escrevendo a imagem do cartão microSD que você salva com este procedimento em um novo cartão microSD e continue os tutoriais dethe section called “Etapa 2: Provisão do Raspberry Pi emAWS IoT” (p. 140).

Para salvar a imagem do cartão microSD em um arquivo:

Na janela do terminal do computador host local que está conectado ao seu Raspberry Pi:

1. Confirme se oConta da AWScredenciais não foram armazenadas.

   a. Execute aAWSConfigurar aplicativo com este comando:

```
aws configure
```

b. Se suas credenciais tiverem sido armazenadas (se elas forem exibidas no prompt), insira o**XYXYXYXYX**String Quando solicitado conforme mostrado aqui. DeixeNome da região padrãoeFormato de saída padrãoEm branco.

```
AWS Access Key ID [****************YXYX]: XYXYXYXYX
AWS Secret Access Key [****************YXYX]: XYXYXYXYX
Default region name:
Default output format:
```

2. Digite este comando para encerrar o Raspberry Pi.

```
sudo shutdown -h 0
```

3. Depois que o Raspberry Pi desligar completamente, remova o conector de alimentação.
4. Remova o cartão microSD do dispositivo.
5. No computador host local:

   a. Insira o cartão microSD.
   b. Usando a ferramenta de imagem do cartão SD, salve a imagem do cartão microSD em um arquivo.
   c. Depois que a imagem do cartão microSD for salva, ejete o cartão do computador host local.

Você pode continuar com este cartão microSD emthe section called “Etapa 2: Provisão do Raspberry Pi emAWS IoT” (p. 140).

Etapa 2: Provisão do Raspberry Pi emAWS IoT

Os procedimentos nesta seção começam com a imagem microSD salva que possui oAWS CLIeAWS IoTDevice Client instalado e crie oAWS IoTrecursos e certificados de dispositivo que provisionam seu Raspberry Pi emAWS IoT.

Instale o cartão microSD no Raspberry Pi

Este procedimento instala o cartão microSD com o software necessário carregado e configurado no Raspberry Pi e configura seuConta da AWSpara que você possa continuar com os tutoriais neste caminho de aprendizado.

Use um cartão microSD dethe section called “(Opcional) Salve a imagem do cartão microSD” (p. 139)que possui o software necessário para os exercícios e tutoriais neste caminho de aprendizagem.

Para instalar o cartão microSD no Raspberry Pi

1. Com a energia desconectada do Raspberry Pi, insira o cartão microSD no Raspberry Pi.
2. Aplique energia ao Raspberry Pi.
3. Após cerca de um minuto, no computador host local, reinicie a sessão da janela do terminal e faça login no Raspberry Pi.
4. Em seu computador host local, na janela do terminal e com oAccess Key IDeSecret Access Key (Chave de acesso secreta)Credenciais do para seu Raspberry Pi:

   a. Execute aAWSConfigurar aplicativo com este comando:

```
aws configure
```

   b. Insira seuConta da AWScredenciais e informações de configuração quando solicitado:

```
AWS Access Key ID [****************YXYX]: your Access Key ID
AWS Secret Access Key [****************YXYX]: your Secret Access Key
Default region name [us-west-2]: your Região da AWS code
Default output format [json]: json
```

Depois de restaurar o seuConta da AWSCredenciais, você está pronto para continuarthe section called “Provisione seu dispositivo emAWS IoT Core” (p. 141).

Provisione seu dispositivo emAWS IoT Core

Os procedimentos nesta seção criam oAWS IoTrecursos que provisionam seu Raspberry Pi emAWS IoT. Ao criar esses recursos, você será solicitado a registrar várias informações. Essas informações são usadas peloAWS IoTConfiguração do Cliente de dispositivo no próximo procedimento.

Para que seu Raspberry Pi trabalhe comAWS IoT, ele deve ser provisionado. O provisionamento é o processo de criação e configuração doAWS IoTRecursos necessários para suportar seu Raspberry Pi como um dispositivo de IoT.

Com o Raspberry Pi ligado e reiniciado, conecte a janela do terminal no computador host local ao Raspberry Pi e conclua esses procedimentos.

Procedimentos nesta seção:

- Criar e baixar arquivos de certificado de dispositivo (p. 141)
- CriarAWS IoTrecursos (p. 142)

Criar e baixar arquivos de certificado de dispositivo

Este procedimento cria os arquivos de certificado do dispositivo para esta demonstração.

Para criar e baixar os arquivos de certificado do dispositivo para seu Raspberry Pi

1. Na janela do terminal do computador host local, insira esses comandos para criar os arquivos de certificado do dispositivo para o dispositivo.

```
mkdir ~/certs/testconn
aws iot create-keys-and-certificate \
--set-as-active \
--certificate-pem-outfile "~/certs/testconn/device.pem.crt" \
--public-key-outfile "~/certs/testconn/public.pem.key" \
--private-key-outfile "~/certs/testconn/private.pem.key"
```

   O comando retorna uma resposta como a seguinte. Registre o*certificateArn*Valor para uso posterior.

```
{
    "certificateArn": "arn:aws:iot:us-
west-2:57EXAMPLE833:cert/76e7e4edb3e52f52334be2f387a06145b2aa4c7fcd810f3aea2d92abc227d269",
    "certificateId":
 "76e7e4edb3e52f5233EXAMPLE7a06145b2aa4c7fcd810f3aea2d92abc227d269",
    "certificatePem": "-----BEGIN CERTIFICATE-----
\nMIIDWTCCAkGgAwIBAgI_SHORTENED_FOR_EXAMPLE_Lgn4jfgtS\n-----END CERTIFICATE-----\n",
    "keyPair": {
        "PublicKey": "-----BEGIN PUBLIC KEY-----
\nMIIBIjANBgkqhkiG9w0BA_SHORTENED_FOR_EXAMPLE_ImwIDAQAB\n-----END PUBLIC KEY-----\n",
        "PrivateKey": "-----BEGIN RSA PRIVATE KEY-----
\nMIIEowIBAAKCAQE_SHORTENED_FOR_EXAMPLE_T9RoDiukY\n-----END RSA PRIVATE KEY-----\n"
```

```
    }
}
```

2. Insira os comandos a seguir para definir as permissões no diretório de certificados e seus arquivos.

```
chmod 745 ~
chmod 700 ~/certs/testconn
chmod 644 ~/certs/testconn/*
chmod 600 ~/certs/testconn/private.pem.key
```

3. Execute este comando para revisar as permissões em seus diretórios e arquivos de certificados.

```
ls -l ~/certs/testconn
```

A saída do comando deve ser a mesma que você vê aqui, exceto que as datas e horas do arquivo serão diferentes.

```
-rw-r--r-- 1 pi pi 1220 Oct 28 13:02 device.pem.crt
-rw------- 1 pi pi 1675 Oct 28 13:02 private.pem.key
-rw-r--r-- 1 pi pi  451 Oct 28 13:02 public.pem.key
```

Neste ponto, você tem os arquivos de certificado do dispositivo instalados em seu Raspberry Pi e você pode continuarthe section called “CriarAWS IoTrecursos” (p. 142).

CriarAWS IoTrecursos

Este procedimento provisiona seu dispositivo emAWS IoTcriando os recursos que seu dispositivo precisa acessarAWS IoTrecursos e serviços.

Para provisionar seu dispositivo emAWS IoT

1. Na janela do terminal do computador host local, insira o seguinte comando para obter o endereço do endpoint de dados do dispositivo para oConta da AWS.

```
aws iot describe-endpoint --endpoint-type IoT:Data-ATS
```

O comando das etapas anteriores retorna uma resposta como a seguinte. Registre o*endpointAddress*Valor para uso posterior.

```
{
    "endpointAddress": "a3qjEXAMPLEffp-ats.iot.us-west-2.amazonaws.com"
}
```

2. Insira este comando para criar umAWS IoTrecurso de coisa do para seu Raspberry Pi.

```
aws iot create-thing --thing-name "DevCliTestThing"
```

Se suas receitasAWS IoTrecurso coisa foi criada, o comando retorna uma resposta como essa.

```
{
    "thingName": "DevCliTestThing",
    "thingArn": "arn:aws:iot:us-west-2:57EXAMPLE833:thing/DevCliTestThing",
    "thingId": "8ea78707-32c3-4f8a-9232-14bEXAMPLEfd"
}
```

3. Na janela do terminal:

a. Abra um editor de texto, como`nano`.

b. Copie este documento de política JSON e cole-o no editor de texto aberto.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:Publish",
                "iot:Subscribe",
                "iot:Receive",
                "iot:Connect"
            ],
            "Resource": [
                "*"
            ]
        }
    ]
}
```

Note

Este documento de política concede generosamente a todos os recursos permissão para se conectar, receber, publicar e assinar. Normalmente, as políticas concedem somente permissão para recursos específicos para executar ações específicas. No entanto, para o teste inicial de conectividade do dispositivo, essa política excessivamente geral e permissiva é usada para minimizar a chance de um problema de acesso durante este teste. Nos tutoriais subsequentes, documentos de políticas com escopo mais estreito serão usados para demonstrar melhores práticas no design de políticas.

c. Salve o arquivo no editor de texto como**`~/policies/dev_cli_test_thing_policy.json`**.

4. Execute este comando para usar o documento de política das etapas anteriores para criar umAWS IoTpolítica.

```
aws iot create-policy \
--policy-name "DevCliTestThingPolicy" \
--policy-document "file://~/policies/dev_cli_test_thing_policy.json"
```

Se a política for criada, o comando retornará uma resposta como essa.

```
{
    "policyName": "DevCliTestThingPolicy",
    "policyArn": "arn:aws:iot:us-west-2:57EXAMPLE833:policy/DevCliTestThingPolicy",
    "policyDocument": "{\n    \"Version\": \"2012-10-17\",\n    \"Statement\": [\n
     {\n            \"Effect\": \"Allow\",\n            \"Action\": [\n
 \"iot:Publish\",\n                \"iot:Subscribe\",\n                \"iot:Receive\",
\n                \"iot:Connect\"\n            ],\n            \"Resource\": [\n
         \"*\"\n            ]\n        }\n    ]\n}\n",
    "policyVersionId": "1"
}
```

5. Execute este comando para anexar a política ao certificado do dispositivo. Substituir*`certificateArn`*com o`certificateArn`valor que você salvou anteriormente.

```
aws iot attach-policy \
--policy-name "DevCliTestThingPolicy" \
--target "certificateArn"
```

Se houver êxito, o comando não retorna nada.

6. Execute este comando para anexar o certificado do dispositivo àAWS IoTrecurso de coisa. Substituir*certificateArn*com o`certificateArn`valor que você salvou anteriormente.

```
aws iot attach-thing-principal \
--thing-name "DevCliTestThing" \
--principal "certificateArn"
```

Se houver êxito, o comando não retorna nada.

Depois de provisionar seu dispositivo com êxito emAWS IoT, você está pronto para continuar athe section called “Etapa 3: Configurar aAWS IoTCliente de dispositivo para testar a conectividade” (p. 144).

Etapa 3: Configurar aAWS IoTCliente de dispositivo para testar a conectividade

Os procedimentos nesta seção configuram oAWS IoTCliente de dispositivo para publicar uma mensagem do MQTT do Raspberry Pi.

Procedimentos nesta seção:

- Criar o arquivo de configuração (p. 144)
- Cliente de teste MQTT aberto (p. 145)
- Execução doAWS IoTCliente do dispositivo (p. 146)

Criar o arquivo de configuração

Este procedimento cria o arquivo de configuração para testar oAWS IoTCliente do dispositivo.

Para criar o arquivo de configuração para testar oAWS IoTCliente do dispositivo

- Na janela do terminal do computador host local que está conectado ao seu Raspberry Pi:

  a. Insira estes comandos para criar um diretório para os arquivos de configuração e definir a permissão no diretório:

  ```
  mkdir ~/dc-configs
  chmod 745 ~/dc-configs
  ```

  b. Abra um editor de texto, como`nano`.

  c. Copie este documento JSON e cole-o no editor de texto aberto.

  ```
  {
    "endpoint": "a3qEXAMPLEaffp-ats.iot.us-west-2.amazonaws.com",
    "cert": "~/certs/testconn/device.pem.crt",
    "key": "~/certs/testconn/private.pem.key",
    "root-ca": "~/certs/AmazonRootCA1.pem",
    "thing-name": "DevCliTestThing",
    "logging": {
      "enable-sdk-logging": true,
      "level": "DEBUG",
      "type": "STDOUT",
      "file": ""
    },
    "jobs": {
  ```

```
    "enabled": false,
    "handler-directory": ""
  },
  "tunneling": {
    "enabled": false
  },
  "device-defender": {
    "enabled": false,
    "interval": 300
  },
  "fleet-provisioning": {
    "enabled": false,
    "template-name": "",
    "template-parameters": "",
    "csr-file": "",
    "device-key": ""
  },
  "samples": {
    "pub-sub": {
      "enabled": true,
      "publish-topic": "test/dc/pubtopic",
      "publish-file": "",
      "subscribe-topic": "test/dc/subtopic",
      "subscribe-file": ""
    }
  },
  "config-shadow": {
    "enabled": false
  },
  "sample-shadow": {
    "enabled": false,
    "shadow-name": "",
    "shadow-input-file": "",
    "shadow-output-file": ""
  }
}
```

d. Substitua o*Endpoint*valor com endpoint de dados do dispositivo para seuConta da AWSque você encontrou emthe section called “Provisione seu dispositivo emAWS IoT Core” (p. 141).

e. Salve o arquivo no editor de texto como**~/dc-configs/dc-testconn-config.json**.

f. Execute este comando para definir as permissões no novo arquivo de configuração.

```
chmod 644 ~/dc-configs/dc-testconn-config.json
```

Depois de salvar o arquivo, você estará pronto para continuarthe section called “Cliente de teste MQTT aberto” (p. 145).

Cliente de teste MQTT aberto

Este procedimento prepara oCliente de teste MQTTnoAWS IoTConsole para assinar a mensagem do MQTT que oAWS IoTO Device Client publica quando ele é executado.

Para preparar oCliente de teste MQTTpara assinar todas as mensagens MQTT

1. No computador host local, noAWS IoTconsole, escolhaCliente de teste MQTT.
2. NoAssinar um tópicoguia, emFiltro de tópicos, insira**#**(um único sinal de libra) e escolhaSubscribepara se inscrever em todos os tópicos do MQTT.
3. Abaixo doAssinaturaslabel, confirme que você vê**#**(um único sinal de libra).

Deixe a janela com oCliente de teste MQTTaberto enquanto você continua athe section called “Execução doAWS IoTCliente do dispositivo” (p. 146).

Execução doAWS IoTCliente do dispositivo

Este procedimento executa oAWS IoTDevice Client para que ele publique uma única mensagem MQTT que oCliente de teste MQTTrecebe e exibe.

Para enviar uma mensagem MQTT doAWS IoTCliente do dispositivo

1. Certifique-se de que tanto a janela do terminal conectada ao seu Raspberry Pi quanto a janela com oCliente de teste MQTTSão visíveis enquanto executa esse procedimento.
2. Na janela do terminal, digite esses comandos para executar oAWS IoTDevice Client usando o arquivo de configuração criado emthe section called “Criar o arquivo de configuração” (p. 144).

   ```
   cd ~/aws-iot-device-client/build
   ./aws-iot-device-client --config-file ~/dc-configs/dc-testconn-config.json
   ```

   Na janela do terminal, oAWS IoTO Device Client exibe mensagens de informações e quaisquer erros que ocorrem quando ele é executado.

   Se nenhum erro for exibido na janela do terminal, consulte oCliente de teste MQTT.
3. NoCliente de teste MQTT, na janela Assinaturas, consulte oHello World!mensagem enviada para `otest/dc/pubtopic`tópico da mensagem.
4. Se oAWS IoTO Device Client não exibe erros e você vêHello World!enviado para `otest/dc/pubtopic`mensagem noCliente de teste MQTT, você demonstrou uma conexão bem-sucedida.
5. Na janela do terminal, digite`^C`(Ctrl-C) para interromper oAWS IoTCliente do dispositivo.

Depois de demonstrar que oAWS IoTO Device Client está funcionando corretamente em seu Raspberry Pi e pode se comunicar comAWS IoT, você pode continuar para othe section called “Demonstrar comunicação de mensagens MQTT com oAWS IoTCliente do dispositivo” (p. 146).

Tutorial: Demonstrar comunicação de mensagens MQTT com oAWS IoTCliente do dispositivo

Este tutorial demonstra como oAWS IoTO Device Client pode assinar e publicar mensagens MQTT, que são comumente usadas em soluções de IoT.

Para iniciar este tutorial:

- Tenha seu computador host local e o Raspberry Pi configurados conforme usado ema seção anterior (p. 137).

  Se você salvou a imagem do cartão microSD após a instalação doAWS IoTDevice Client, você pode usar um cartão microSD com essa imagem com seu Raspberry Pi.
- Se você já executou esta demonstração antes, revise??? (p. 171)Para excluir todos osAWS IoTrecursos que você criou em execuções anteriores para evitar erros de recursos duplicados.

A conclusão deste tutorial requer cerca de 45 minutos.

Quando terminar este tópico:

- Você demonstrará diferentes maneiras pelas quais seu dispositivo IoT pode assinar mensagens MQTT deAWS IoTe publique mensagens MQTT paraAWS IoT.

Equipamento necessário:

- Seu ambiente local de desenvolvimento e teste dea seção anterior (p. 137)
- O Raspberry Pi que você usou ema seção anterior (p. 137)
- O cartão de memória microSD do Raspberry Pi que você usou ema seção anterior (p. 137)

Procedimentos neste tutorial

- Etapa 1: Preparar o Raspberry Pi para demonstrar a comunicação de mensagens MQTT (p. 147)
- Etapa 2: Demonstrar mensagens de publicação com oAWS IoTCliente do dispositivo (p. 152)
- Etapa 3: Demonstrar a assinatura de mensagens com oAWS IoTCliente do dispositivo (p. 155)

Etapa 1: Preparar o Raspberry Pi para demonstrar a comunicação de mensagens MQTT

Este procedimento cria os recursos emAWS IoTe no Raspberry Pi para demonstrar comunicação de mensagens MQTT usando oAWS IoTCliente do dispositivo.

Procedimentos nesta seção:

- Crie os arquivos de certificado para demonstrar a comunicação MQTT (p. 147)
- Provisione seu dispositivo para demonstrar comunicação MQTT (p. 148)
- Configurar aAWS IoTArquivo de configuração do Device Client e cliente de teste MQTT para demonstrar comunicação MQTT (p. 150)

Crie os arquivos de certificado para demonstrar a comunicação MQTT

Este procedimento cria os arquivos de certificado do dispositivo para esta demonstração.

Para criar e baixar os arquivos de certificado do dispositivo para seu Raspberry Pi

1. Na janela do terminal do computador host local, insira o seguinte comando para criar os arquivos de certificado do dispositivo para o dispositivo.

```
mkdir ~/certs/pubsub
aws iot create-keys-and-certificate \
--set-as-active \
--certificate-pem-outfile "~/certs/pubsub/device.pem.crt" \
--public-key-outfile "~/certs/pubsub/public.pem.key" \
--private-key-outfile "~/certs/pubsub/private.pem.key"
```

   O comando retornará uma resposta como a seguinte. Salvar o*certificateArn*valor para uso posterior.

```
{
"certificateArn": "arn:aws:iot:us-
west-2:57EXAMPLE833:cert/76e7e4edb3e52f52334be2f387a06145b2aa4c7fcd810f3aea2d92abc227d269",
"certificateId": "76e7e4edb3e52f5233EXAMPLE7a06145b2aa4c7fcd810f3aea2d92abc227d269",
"certificatePem": "-----BEGIN CERTIFICATE-----
\nMIIDWTCCAkGgAwIBAgI_SHORTENED_FOR_EXAMPLE_Lgn4jfgtS\n-----END CERTIFICATE-----\n",
"keyPair": {
    "PublicKey": "-----BEGIN PUBLIC KEY-----
\nMIIBIjANBgkqhkiG9w0BA_SHORTENED_FOR_EXAMPLE_ImwIDAQAB\n-----END PUBLIC KEY-----\n",
```

```
    "PrivateKey": "-----BEGIN RSA PRIVATE KEY-----
\nMIIEowIBAAKCAQE_SHORTENED_FOR_EXAMPLE_T9RoDiukY\n-----END RSA PRIVATE KEY-----\n"
}
}
```

2. Insira os comandos a seguir para definir as permissões no diretório de certificados e seus arquivos.

```
chmod 700 ~/certs/pubsub
chmod 644 ~/certs/pubsub/*
chmod 600 ~/certs/pubsub/private.pem.key
```

3. Execute este comando para revisar as permissões em seus diretórios e arquivos de certificados.

```
ls -l ~/certs/pubsub
```

A saída do comando deve ser a mesma que você vê aqui, exceto que as datas e horas do arquivo serão diferentes.

```
-rw-r--r-- 1 pi pi 1220 Oct 28 13:02 device.pem.crt
-rw------- 1 pi pi 1675 Oct 28 13:02 private.pem.key
-rw-r--r-- 1 pi pi  451 Oct 28 13:02 public.pem.key
```

4. Insira esses comandos para criar os diretórios para os arquivos de log.

```
mkdir ~/.aws-iot-device-client
mkdir ~/.aws-iot-device-client/log
chmod 745 ~/.aws-iot-device-client/log
echo " " > ~/.aws-iot-device-client/log/aws-iot-device-client.log
echo " " > ~/.aws-iot-device-client/log/pubsub_rx_msgs.log
chmod 600 ~/.aws-iot-device-client/log/*
```

Provisione seu dispositivo para demonstrar comunicação MQTT

Esta seção cria oAWS IoTrecursos que provisionam seu Raspberry Pi emAWS IoT.

Para provisionar seu dispositivo emAWS IoT:

1. Na janela do terminal do computador host local, insira o seguinte comando para obter o endereço do endpoint de dados do dispositivo para oConta da AWS.

```
aws iot describe-endpoint --endpoint-type IoT:Data-ATS
```

O valor do endpoint não mudou desde o momento em que você executou este comando para o tutorial anterior. Executar o comando novamente aqui é feito para facilitar a localização e colar o valor do endpoint de dados no arquivo de configuração usado neste tutorial.

O comando das etapas anteriores retornará uma resposta como a seguinte. Grave o*endpointAddress*valor para uso posterior.

```
{
"endpointAddress": "a3qjEXAMPLEffp-ats.iot.us-west-2.amazonaws.com"
}
```

2. Digite este comando para criar um novoAWS IoTrecurso de coisa para seu Raspberry Pi.

```
aws iot create-thing --thing-name "PubSubTestThing"
```

Como umAWS IoTrecurso coisa é umvirtualrepresentação do seu dispositivo na nuvem, podemos criar vários recursos emAWS IoTPara usar para diferentes propósitos. Todos eles podem ser usados pelo mesmo dispositivo de IoT físico para representar diferentes aspectos do dispositivo.

Esses tutoriais usarão apenas um recurso de cada vez para representar o Raspberry Pi. Dessa forma, nesses tutoriais, eles representam as diferentes demonstrações para que depois de criar oAWS IoTrecursos para uma demonstração, você pode voltar e repetir a demonstração usando os recursos criados especificamente para cada um.

Se suas receitasAWS IoTA coisa do recurso foi criada, o comando retornará uma resposta como esta.

```
{
"thingName": "PubSubTestThing",
"thingArn": "arn:aws:iot:us-west-2:57EXAMPLE833:thing/PubSubTestThing",
"thingId": "8ea78707-32c3-4f8a-9232-14bEXAMPLEfd"
}
```

3. Na janela do terminal:

   a. Abra um editor de texto, como`nano`.

   b. Copie este documento JSON e cole-o no editor de texto aberto.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iot:Connect"
      ],
      "Resource": [
        "arn:aws:iot:us-west-2:57EXAMPLE833:client/PubSubTestThing"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "iot:Publish"
      ],
      "Resource": [
        "arn:aws:iot:us-west-2:57EXAMPLE833:topic/test/dc/pubtopic"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "iot:Subscribe"
      ],
      "Resource": [
        "arn:aws:iot:us-west-2:57EXAMPLE833:topicfilter/test/dc/subtopic"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "iot:Receive"
      ],
      "Resource": [
        "arn:aws:iot:us-west-2:57EXAMPLE833:topic/test/dc/subtopic"
      ]
    }
  ]
```

```
}
```

c. No editor, em cada`Resource`seção do documento de política, substitua*us-west- 2:57 EXEMPLO833*com o seuRegião da AWS, um caractere de dois pontos (:) e seus 12 dígitosConta da AWSnúmero.

d. Salve o arquivo no editor de texto como**~/policies/pubsub_test_thing_policy.json**.

4. Execute este comando para usar o documento de política das etapas anteriores para criar umAWS IoTpolítica.

```
aws iot create-policy \
--policy-name "PubSubTestThingPolicy" \
--policy-document "file://~/policies/pubsub_test_thing_policy.json"
```

Se a política for criada, o comando retornará uma resposta como esta.

```
{
    "policyName": "PubSubTestThingPolicy",
    "policyArn": "arn:aws:iot:us-west-2:57EXAMPLE833:policy/PubSubTestThingPolicy",
    "policyDocument": "{\n\"Version\": \"2012-10-17\",\n\"Statement\": [\n{\n
\"Effect\": \"Allow\",\n\"Action\": [\n\"iot:Connect\"\n],\n\"Resource\": [\n
\"arn:aws:iot:us-west-2:57EXAMPLE833:client/PubSubTestThing\"\n]\n},\n{\n\"Effect\":
 \"Allow\",\n\"Action\": [\n\"iot:Publish\"\n],\n\"Resource\": [\n\"arn:aws:iot:us-
west-2:57EXAMPLE833:topic/test/dc/pubtopic\"\n]\n},\n{\n\"Effect\": \"Allow
\",\n\"Action\": [\n\"iot:Subscribe\"\n],\n\"Resource\": [\n\"arn:aws:iot:us-
west-2:57EXAMPLE833:topicfilter/test/dc/subtopic\"\n]\n},\n{\n\"Effect\": \"Allow
\",\n\"Action\": [\n\"iot:Receive\"\n],\n\"Resource\": [\n\"arn:aws:iot:us-
west-2:57EXAMPLE833:topic/test/dc/*\"\n]\n}\n]\n}\n",
    "policyVersionId": "1"
```

5. Execute este comando para anexar a política ao certificado do dispositivo. Substituir*certificateArn*com o`certificateArn`O valor que você salvou anteriormente nesta seção.

```
aws iot attach-policy \
--policy-name "PubSubTestThingPolicy" \
--target "certificateArn"
```

Se houver êxito, o comando não retorna nada.

6. Execute este comando para anexar o certificado do dispositivo aoAWS IoTrecurso de coisa. Substituir*certificateArn*com o`certificateArn`O valor que você salvou anteriormente nesta seção.

```
aws iot attach-thing-principal \
--thing-name "PubSubTestThing" \
--principal "certificateArn"
```

Se houver êxito, o comando não retorna nada.

Depois de provisionar seu dispositivo com êxito emAWS IoT, você está pronto para continuarthe section called “Configurar aAWS IoTArquivo de configuração do Device Client e cliente de teste MQTT para demonstrar comunicação MQTT” (p. 150).

Configurar aAWS IoTArquivo de configuração do Device Client e cliente de teste MQTT para demonstrar comunicação MQTT

Este procedimento cria um arquivo de configuração para testar oAWS IoTCliente do dispositivo.

Para criar o arquivo de configuração para testar oAWS IoTCliente do dispositivo

1. Na janela do terminal do computador host local que está conectado ao Raspberry Pi:

   a. Abra um editor de texto, como`nano`.

   b. Copie este documento JSON e cole-o no editor de texto aberto.

```
{
  "endpoint": "a3qEXAMPLEaffp-ats.iot.us-west-2.amazonaws.com",
  "cert": "~/certs/pubsub/device.pem.crt",
  "key": "~/certs/pubsub/private.pem.key",
  "root-ca": "~/certs/AmazonRootCA1.pem",
  "thing-name": "PubSubTestThing",
  "logging": {
    "enable-sdk-logging": true,
    "level": "DEBUG",
    "type": "STDOUT",
    "file": ""
  },
  "jobs": {
    "enabled": false,
    "handler-directory": ""
  },
  "tunneling": {
    "enabled": false
  },
  "device-defender": {
    "enabled": false,
    "interval": 300
  },
  "fleet-provisioning": {
    "enabled": false,
    "template-name": "",
    "template-parameters": "",
    "csr-file": "",
    "device-key": ""
  },
  "samples": {
    "pub-sub": {
      "enabled": true,
      "publish-topic": "test/dc/pubtopic",
      "publish-file": "",
      "subscribe-topic": "test/dc/subtopic",
      "subscribe-file": "~/.aws-iot-device-client/log/pubsub_rx_msgs.log"
    }
  },
  "config-shadow": {
    "enabled": false
  },
  "sample-shadow": {
    "enabled": false,
    "shadow-name": "",
    "shadow-input-file": "",
    "shadow-output-file": ""
  }
}
```

   c. Substitua o*`Endpoint`*valor com endpoint de dados do dispositivo para seuConta da AWSque você encontrou emthe section called “Provisione seu dispositivo emAWS IoT Core” (p. 141).

   d. Salve o arquivo no editor de texto como**`~/dc-configs/dc-pubsub-config.json`**.

   e. Execute este comando para definir as permissões no novo arquivo de configuração.

```
chmod 644 ~/dc-configs/dc-pubsub-config.json
```

2. Para preparar oCliente de teste MQTT dapara assinar todas as mensagens MQTT:

   a. No computador host local, noAWS IoTconsole, escolhaCliente de teste MQTT da.
   b. NoAssinar um tópicoGuia doFiltro de tópicos, insira**#**(um único sinal de libra) e escolhaSubscribe.
   c. Abaixo doAssinaturaslabel, confirme que você vê**#**(um único sinal de libra).

   Deixe a janela com oCliente de teste MQTT daabra enquanto você continua este tutorial.

Depois de salvar o arquivo e configurar oCliente de teste MQTT da, você está pronto para continuarthe section called “Etapa 2: Demonstrar mensagens de publicação com oAWS IoTCliente do dispositivo” (p. 152).

Etapa 2: Demonstrar mensagens de publicação com oAWS IoTCliente do dispositivo

Os procedimentos nesta seção demonstram como oAWS IoTO Device Client pode enviar mensagens MQTT padrão e personalizadas.

Essas declarações de política na política que você criou na etapa anterior para esses exercícios dão ao Raspberry Pi permissão para executar estas ações:

- **`iot:Connect`**

  Dá ao cliente chamado`PubSubTestThing`, seu Raspberry Pi executando oAWS IoTDevice Client, para se conectar.

```
{
  "Effect": "Allow",
  "Action": [
    "iot:Connect"
  ],
  "Resource": [
    "arn:aws:iot:us-west-2:57EXAMPLE833:client/PubSubTestThing"
  ]
}
```

- **`iot:Publish`**

  Dá ao Raspberry Pi permissão para publicar mensagens com um tópico MQTT de`test/dc/pubtopic`.

```
{
  "Effect": "Allow",
  "Action": [
    "iot:Publish"
  ],
  "Resource": [
    "arn:aws:iot:us-west-2:57EXAMPLE833:topic/test/dc/pubtopic"
  ]
}
```

  O`iot:Publish`a ação dá permissão para publicar nos tópicos do MQTT listados na matriz Resource. Oconteúdodessas mensagens não é controlada pela declaração de política.

Publique a mensagem padrão usando oAWS IoTCliente do dispositivo

Este procedimento executa oAWS IoTDevice Client para que ele publique uma única mensagem MQTT padrão que oCliente de teste MQTT darecebe e exibe.

Para enviar a mensagem MQTT padrão doAWS IoTCliente do dispositivo

1. Certifique-se de que a janela do terminal no computador host local está conectada ao Raspberry Pi e a janela com oCliente de teste MQTT daSão visíveis enquanto você executa este procedimento.
2. Na janela do terminal, digite esses comandos para executar aAWS IoTDevice Client usando o arquivo de configuração criado emthe section called “Criar o arquivo de configuração” (p. 144).

   ```
   cd ~/aws-iot-device-client/build
   ./aws-iot-device-client --config-file ~/dc-configs/dc-pubsub-config.json
   ```

   Na janela do terminal, oAWS IoTO Device Client exibe mensagens de informações e quaisquer erros que ocorrem quando ele é executado.

   Se nenhum erro for exibido na janela do terminal, consulte aCliente de teste MQTT da.
3. NoCliente de teste MQTT da, noAssinaturasjanela, veja oHello World!mensagem enviada para `otest/dc/pubtopic`tópico da mensagem.
4. Se oAWS IoTO Device Client não exibe erros e você vêHello World!enviado para `otest/dc/pubtopic`mensagem noCliente de teste MQTT da, você demonstrou uma conexão bem-sucedida.
5. Na janela do terminal, digite**`^C`**(Ctrl-C) para interromper oAWS IoTCliente do dispositivo.

Depois de demonstrar que oAWS IoTDevice Client publicou a mensagem MQTT padrão, você pode continuar com othe section called “Publique uma mensagem personalizada usando oAWS IoTCliente do dispositivo.” (p. 153).

Publique uma mensagem personalizada usando oAWS IoTCliente do dispositivo.

Os procedimentos nesta seção criam uma mensagem MQTT personalizada e, em seguida, executam oAWS IoTDevice Client para que ele publique a mensagem MQTT personalizada uma vez para oCliente de teste MQTT dapara receber e exibir.

Criar uma mensagem MQTT personalizada para oAWS IoTCliente do dispositivo

Execute estas etapas na janela do terminal no computador host local conectado ao seu Raspberry Pi.

Para criar uma mensagem personalizada para oAWS IoTDevice Client para publicar

1. Na janela do terminal, abra um editor de texto, como`nano`.
2. No editor de texto, copie e cole o documento JSON a seguir. Esta será a carga útil da mensagem MQTT que oAWS IoTO Device Client publica.

   ```
   {
     "temperature": 28,
     "humidity": 80,
     "barometer": 1013,
     "wind": {
       "velocity": 22,
       "bearing": 255
     }
   }
   ```
3. Salve o conteúdo do editor de texto como**`~/messages/sample-ws-message.json`**.

4. Digite o comando a seguir para definir as permissões do arquivo de mensagem que você acabou de criar.

```
chmod 600 ~/messages/*
```

Para criar um arquivo de configuração para oAWS IoTCliente de dispositivo a ser usado para enviar a mensagem personalizada

1. Na janela do terminal, em um editor de texto como`nano`, abra o existenteAWS IoTArquivo de configuração do Device Client:**`~/dc-configs/dc-pubsub-config.json`**.
2. Edite o`samples`seja parecido com isto. Nenhuma outra parte desse arquivo precisa ser alterada.

```
  "samples": {
    "pub-sub": {
      "enabled": true,
      "publish-topic": "test/dc/pubtopic",
      "publish-file": "~/messages/sample-ws-message.json",
      "subscribe-topic": "test/dc/subtopic",
      "subscribe-file": "~/.aws-iot-device-client/log/pubsub_rx_msgs.log"
```

3. Salve o conteúdo do editor de texto como**`~/dc-configs/dc-pubsub-custom-config.json`**.
4. Execute este comando para definir as permissões no novo arquivo de configuração.

```
chmod 644 ~/dc-configs/dc-pubsub-custom-config.json
```

Publique a mensagem MQTT personalizada usando oAWS IoTCliente do dispositivo

Essa alteração afeta somente oconteúdosda carga útil da mensagem MQTT, portanto, a política atual continuará funcionando. No entanto, se oTópico MQTT do(conforme definido pelo`publish-topic`valor em`~/dc-configs/dc-pubsub-custom-config.json`) foi alterado, `oiot::Publish`declaração política também precisaria ser modificada para permitir que o Raspberry Pi publique no novo tópico MQTT.

Para enviar a mensagem MQTT doAWS IoTCliente do dispositivo

1. Certifique-se de que tanto a janela do terminal quanto a janela com oCliente de teste MQTT daSão visíveis enquanto você executa este procedimento. Além disso, verifique se oCliente de teste MQTT daainda está inscrito no#Filtro de tópicos. Se não estiver, assine o#filtro de tópico novamente.
2. Na janela do terminal, digite esses comandos para executar aAWS IoTDevice Client usando o arquivo de configuração criado emthe section called “Criar o arquivo de configuração” (p. 144).

```
cd ~/aws-iot-device-client/build
./aws-iot-device-client --config-file ~/dc-configs/dc-pubsub-custom-config.json
```

   Na janela do terminal, oAWS IoTO Device Client exibe mensagens de informações e quaisquer erros que ocorrem quando ele é executado.

   Se nenhum erro for exibido na janela do terminal, revise o cliente de teste MQTT.
3. NoCliente de teste MQTT da, noAssinaturas, consulte a carga útil da mensagem personalizada enviada para `otest/dc/pubtopic`tópico da mensagem.
4. Se oAWS IoTO Device Client não exibe erros e você vê a carga útil da mensagem personalizada publicada na`test/dc/pubtopic`mensagem noCliente de teste MQTT da, você publicou uma mensagem personalizada com sucesso.
5. Na janela do terminal, digite**`^C`**(Ctrl-C) para interromper oAWS IoTCliente do dispositivo.

Depois de demonstrar que oAWS IoTDevice Client publicou uma carga de mensagem personalizada, você pode continuar athe section called “Etapa 3: Demonstrar a assinatura de mensagens com oAWS IoTCliente do dispositivo” (p. 155).

Etapa 3: Demonstrar a assinatura de mensagens com oAWS IoTCliente do dispositivo

Nesta seção, você demonstrará dois tipos de assinaturas de mensagens:

- Assinatura de tópico único
- Assinatura de tópicos Wild-card

Essas declarações políticas na política criada para esses exercícios dão ao Raspberry Pi permissão para executar estas ações:

- **`iot:Receive`**

  Fornece oAWS IoTPermissão do Device Client para receber tópicos do MQTT que correspondem àqueles nomeados na`Resource`objeto.

```
{
  "Effect": "Allow",
  "Action": [
    "iot:Receive"
  ],
  "Resource": [
    "arn:aws:iot:us-west-2:57EXAMPLE833:topic/test/dc/subtopic"
  ]
}
```

- **`iot:Subscribe`**

  Fornece oAWS IoTPermissão do Device Client para assinar filtros de tópico MQTT que correspondem àqueles nomeados na`Resource`objeto.

```
{
  "Effect": "Allow",
  "Action": [
    "iot:Subscribe"
  ],
  "Resource": [
    "arn:aws:iot:us-west-2:57EXAMPLE833:topicfilter/test/dc/subtopic"
  ]
}
```

Assine um único tópico de mensagem MQTT

Este procedimento demonstra como oAWS IoTO Device Client pode assinar e registrar mensagens MQTT.

Na janela do terminal do computador host local conectado ao Raspberry Pi, liste o conteúdo do**`~/dc-configs/dc-pubsub-custom-config.json`**ou abra o arquivo em um editor de texto para revisar seu conteúdo. Localize o`samples`Object, que deve ser semelhante a este.

```
"samples": {
  "pub-sub": {
    "enabled": true,
```

```
    "publish-topic": "test/dc/pubtopic",
    "publish-file": "~/messages/sample-ws-message.json",
    "subscribe-topic": "test/dc/subtopic",
    "subscribe-file": "~/.aws-iot-device-client/log/pubsub_rx_msgs.log"
```

Observe o`subscribe-topic`value é o tópico MQTT para o qual oAWS IoTO Device Client se inscreverá quando for executado. OAWS IoTDevice Client grava as cargas úteis de mensagens recebidas dessa assinatura para o arquivo nomeado na`subscribe-file`value.

Para inscrever-se em um tópico de mensagem MQTT daAWS IoTCliente do dispositivo

1. Certifique-se de que tanto a janela do terminal quanto a janela com o cliente de teste MQTT estejam visíveis enquanto você executa este procedimento. Além disso, verifique se oCliente de teste MQTT daainda está inscrito no#Filtro de tópicos. Se não estiver, assine o#filtro de tópico novamente.
2. Na janela do terminal, digite esses comandos para executar aAWS IoTDevice Client usando o arquivo de configuração criado emthe section called “Criar o arquivo de configuração” (p. 144).

   ```
   cd ~/aws-iot-device-client/build
   ./aws-iot-device-client --config-file ~/dc-configs/dc-pubsub-custom-config.json
   ```

   Na janela do terminal, oAWS IoTO Device Client exibe mensagens de informações e quaisquer erros que ocorrem quando ele é executado.

   Se nenhum erro for exibido na janela do terminal, continue naAWS IoTconsole do .
3. NoAWS IoTconsole doCliente de teste MQTT da, escolha oPublicar em um tópicoGuia.
4. DentroNome do tópico, insira**`test/dc/subtopic`**
5. DentroCarga útil da mensagem, revise o conteúdo da mensagem.
6. SelecionePublicarpara publicar a mensagem MQTT.
7. Na janela do terminal, observe omensagem recebidaEntrada doAWS IoTCliente de dispositivo que será semelhante a esta.

   ```
   2021-11-10T16:02:20.890Z [DEBUG] {samples/PubSubFeature.cpp}: Message received on
    subscribe topic, size: 45 bytes
   ```
8. Depois de ver omensagem recebidaentrada que mostra que a mensagem foi recebida, insira**`^C`**(Ctrl-C) para interromper oAWS IoTCliente do dispositivo.
9. Digite este comando para exibir o final do arquivo de log de mensagens e ver a mensagem que você publicou a partir doCliente de teste MQTT da.

   ```
   tail ~/.aws-iot-device-client/log/pubsub_rx_msgs.log
   ```

Ao visualizar a mensagem no arquivo de log, você demonstrou que oAWS IoTO Device Client recebeu a mensagem que você publicou do cliente de teste MQTT.

Assine vários tópicos de mensagem MQTT usando caracteres curinga

Esses procedimentos demonstram como oAWS IoTO Device Client pode assinar e registrar mensagens MQTT usando caracteres curinga. Para fazer isso, você irá:

1. Atualize o filtro de tópicos doAWS IoTO Device Client usa para assinar tópicos do MQTT.
2. Atualize a política usada pelo dispositivo para permitir as novas assinaturas.
3. Execute aAWS IoTDevice Client e publique mensagens do console de teste MQTT.

Para criar um arquivo de configuração para assinar vários tópicos de mensagens MQTT usando um filtro de tópico curinga MQTT

1. Na janela do terminal do computador host local conectado ao seu Raspberry Pi, abra**~/dc-configs/dc-pubsub-custom-config.json**para editar e localizar `osamples`objeto.
2. No editor de texto, localize `osamples`objeto e atualize `osubscribe-topic`valor para ser parecido com.

```
"samples": {
  "pub-sub": {
    "enabled": true,
    "publish-topic": "test/dc/pubtopic",
    "publish-file": "~/messages/sample-ws-message.json",
    "subscribe-topic": "test/dc/#",
    "subscribe-file": "~/.aws-iot-device-client/log/pubsub_rx_msgs.log"
```

   O novo`subscribe-topic`O valor é umFiltro de tópicos MQTT da (p. 100)com um caractere curinga MQTT no final. Isso descreve uma assinatura de todos os tópicos do MQTT que começam com`test/dc/`. OAWS IoTDevice Client grava as cargas úteis de mensagens recebidas dessa assinatura para o arquivo nomeado em`subscribe-file`.
3. Salve o arquivo de configuração modificado como**~/dc-configs/dc-pubsub-wild-config.json**e saia do editor.

Para modificar a política usada pelo seu Raspberry Pi para permitir a assinatura e o recebimento de vários tópicos de mensagens MQTT

1. Na janela do terminal do seu computador host local conectado ao seu Raspberry Pi, em seu editor de texto favorito, abra**~/policies/pubsub_test_thing_policy.json**para edição e, em seguida, localize o`iot::Subscribe`e`iot::Receive`declarações de política no arquivo.
2. No`iot::Subscribe`declaração de política, atualize a string no objeto Resource para substituir`subtopic`com`*`Para que fique semelhante a isto.

```
{
  "Effect": "Allow",
  "Action": [
    "iot:Subscribe"
  ],
  "Resource": [
    "arn:aws:iot:us-west-2:57EXAMPLE833:topicfilter/test/dc/*"
  ]
}
```

   Note

   OCaracteres curinga do filtro de tópico MQTT (p. 100)são as`+`(sinal de adição) e o`#`(sinal de libra). Uma solicitação de assinatura com um`#`no final se inscreve em todos os tópicos que começam com a string que precede o`#`caractere (por exemplo,`test/dc/`neste caso). O valor do recurso na declaração de política que autoriza esta assinatura, no entanto, deve usar um`*`(um asterisco) no lugar do`#`(um sinal de libra) no filtro de tópico ARN. Isso ocorre porque o processador de políticas usa um caractere curinga diferente do que o MQTT usa. Para obter mais informações sobre o uso de caracteres curinga para tópicos e filtros de tópicos em políticas, consultePolíticas para clientes MQTT (p. 351).
3. No`iot::Receive`declaração de política, atualize a string no objeto Resource para substituir`subtopic`com`*`Para que fique semelhante a isto.

```
{
```

```
      "Effect": "Allow",
      "Action": [
        "iot:Receive"
      ],
      "Resource": [
        "arn:aws:iot:us-west-2:57EXAMPLE833:topic/test/dc/*"
      ]
    }
```

4. Salve o documento de política atualizado como**~/policies/pubsub_wild_test_thing_policy.jsonn**e saia do editor.
5. Digite este comando para atualizar a política deste tutorial para usar as novas definições de recurso.

```
aws iot create-policy-version \
--set-as-default \
--policy-name "PubSubTestThingPolicy" \
--policy-document "file://~/policies/pubsub_wild_test_thing_policy.json"
```

Se o comando for bem-sucedido, ele retornará uma resposta como esta. Note que`policyVersionId`é agora`2`, indicando que esta é a segunda versão desta política.

Se atualizar a política com êxito, você poderá continuar para o próximo procedimento.

```
{
    "policyArn": "arn:aws:iot:us-west-2:57EXAMPLE833:policy/PubSubTestThingPolicy",
    "policyDocument": "{\n  \"Version\": \"2012-10-17\",\n  \"Statement\": [\n    {\n
    \"Effect\": \"Allow\",\n      \"Action\": [\n        \"iot:Connect\"\n      ],\n
   \"Resource\": [\n        \"arn:aws:iot:us-west-2:57EXAMPLE833:client/PubSubTestThing
\"\n      ]\n    },\n    {\n      \"Effect\": \"Allow\",\n      \"Action\": [\n
      \"iot:Publish\"\n      ],\n      \"Resource\": [\n        \"arn:aws:iot:us-
west-2:57EXAMPLE833:topic/test/dc/pubtopic\"\n      ]\n    },\n    {\n      \"Effect
\": \"Allow\",\n      \"Action\": [\n        \"iot:Subscribe\"\n      ],\n
 \"Resource\": [\n        \"arn:aws:iot:us-west-2:57EXAMPLE833:topicfilter/test/dc/
*\"\n      ]\n    },\n    {\n      \"Effect\": \"Allow\",\n      \"Action\": [\n
       \"iot:Receive\"\n      ],\n      \"Resource\": [\n        \"arn:aws:iot:us-
west-2:57EXAMPLE833:topic/test/dc/*\"\n      ]\n    }\n  ]\n}\n",
    "policyVersionId": "2",
    "isDefaultVersion": true
}
```

Se você receber um erro de que há muitas versões de política para salvar uma nova, digite este comando para listar as versões atuais da política. Revise a lista que esse comando retorna para encontrar uma versão de política que você possa excluir.

```
aws iot list-policy-versions --policy-name "PubSubTestThingPolicy"
```

Digite este comando para excluir uma versão da qual você não precisa mais. Observe que você não pode excluir a versão da política padrão. A versão da política padrão é aquela com um`isDefaultVersion`valor de`true`.

```
aws iot delete-policy-version \
--policy-name "PubSubTestThingPolicy" \
--policy-version-id policyId
```

Depois de excluir uma versão de política, tente novamente esta etapa.

Com o arquivo de configuração e a política atualizados, você está pronto para demonstrar assinaturas curinga com oAWS IoTCliente do dispositivo.

Para demonstrar como oAWS IoTO Device Client se inscreve e recebe vários tópicos de mensagens MQTT

1. NoCliente de teste MQTT da, verifique as assinaturas. Se oCliente de teste MQTT daestá inscrito no no no**#**Filtro de tópicos, continue para a próxima etapa. Caso contrário, noCliente de teste MQTT da, emAssinar um tópicoGuia doFiltro de tópicos, insira**#**(um caractere de sinal de libra) e, em seguida, escolhaSubscribepara se inscrever nele.
2. Na janela do terminal do computador host local que está conectado ao Raspberry Pi, insira esses comandos para iniciar oAWS IoTCliente do dispositivo.

```
cd ~/aws-iot-device-client/build
./aws-iot-device-client --config-file ~/dc-configs/dc-pubsub-wild-config.json
```

3. Enquanto assiste aoAWS IoTSaída do Device Client na janela do terminal no computador host local, retorne aoCliente de teste MQTT da. NoPublicar em um tópicoGuia doNome do tópico, insira**`test/dc/subtopic`**e, depois, escolhaPublicar.
4. Na janela do terminal, confirme se a mensagem foi recebida procurando por uma mensagem como:

```
2021-11-10T16:34:20.101Z [DEBUG] {samples/PubSubFeature.cpp}: Message received on
 subscribe topic, size: 76 bytes
```

5. Enquanto assiste aoAWS IoTSaída do Device Client na janela do terminal do computador host local, retorne aoCliente de teste MQTT do. NoPublicar em um tópicoGuia doNome do tópico, insira**`test/dc/subtopic2`**e, depois, escolhaPublicar.
6. Na janela do terminal, confirme se a mensagem foi recebida procurando por uma mensagem como:

```
2021-11-10T16:34:32.078Z [DEBUG] {samples/PubSubFeature.cpp}: Message received on
 subscribe topic, size: 77 bytes
```

7. Depois de ver as mensagens que confirmam que ambas as mensagens foram recebidas, insira**`^C`**(Ctrl-C) para interromper oAWS IoTCliente do dispositivo.
8. Digite este comando para exibir o final do arquivo de log de mensagens e ver a mensagem que você publicou a partir doCliente de teste MQTT do.

```
tail -n 20 ~/.aws-iot-device-client/log/pubsub_rx_msgs.log
```

> **Note**
>
> O arquivo de log contém somente cargas de mensagens. Os tópicos da mensagem não são registrados no arquivo de log de mensagens recebido.
> Você também pode ver a mensagem publicada pelaAWS IoTDevice Client no log recebido. Isso ocorre porque o filtro de tópicos curinga inclui esse tópico de mensagem e, às vezes, a solicitação de assinatura pode ser processada pelo corretor de mensagens antes que a mensagem publicada seja enviada aos assinantes.

As entradas no arquivo de log demonstram que as mensagens foram recebidas. Você pode repetir esse procedimento usando outros nomes de tópicos. Todas as mensagens que têm um nome de tópico que começa com`test/dc/`deve ser recebido e registrado. Mensagens com nomes de tópicos que começam com qualquer outro texto são ignoradas.

Depois de demonstrar como oAWS IoTO Device Client pode publicar e assinar mensagens MQTT, continuar aTutorial: Demonstrar ações remotas (trabalhos) com oAWS IoTCliente do dispositivo (p. 160).

Tutorial: Demonstrar ações remotas (trabalhos) com oAWS IoTCliente do dispositivo

Nesses tutoriais, você configurará e implantará trabalhos em seu Raspberry Pi para demonstrar como você pode enviar operações remotas para seus dispositivos IoT.

Para iniciar este tutorial:

- Tenha seu computador host local um Raspberry Pi configurado conforme usado emA seção anterior (p. 146).
- Se você ainda não concluiu o tutorial na seção anterior, você pode tentar este tutorial usando o Raspberry Pi com um cartão microSD que tem a imagem que você salvou depois de instalar oAWS IoTCliente do dispositivo(Opcional) Salve a imagem do cartão microSD (p. 139).
- Se você já executou esta demonstração antes, revise??? (p. 171)Para excluir todosAWS IoTrecursos que você criou em execuções anteriores para evitar erros de recursos duplicados.

A conclusão deste tutorial requer cerca de 45 minutos.

Quando terminar este tópico:

- Você demonstrará diferentes maneiras pelas quais seu dispositivo IoT pode usar oAWS IoT Corepara executar operações remotas gerenciadas porAWS IoT.

Equipamento necessário:

- Seu ambiente local de desenvolvimento e teste em que você testouuma seção anterior (p. 137)
- O Raspberry Pi em que você testouuma seção anterior (p. 137)
- O cartão de memória microSD do Raspberry Pi que você testouuma seção anterior (p. 137)

Procedimentos neste tutorial

- Etapa 1: Preparar o Raspberry Pi para executar trabalhos (p. 160)
- Etapa 2: Criar e executar o trabalho noAWS IoT (p. 166)

Etapa 1: Preparar o Raspberry Pi para executar trabalhos

Os procedimentos nesta seção descrevem como preparar seu Raspberry Pi para executar trabalhos usando oAWS IoTCliente do dispositivo.

> **Note**
>
> Esses procedimentos são específicos do dispositivo. Se você quiser executar os procedimentos nesta seção com mais de um dispositivo ao mesmo tempo, cada dispositivo precisará de sua própria política e certificado exclusivo e nome específico do dispositivo. Para fornecer a cada dispositivo seus recursos exclusivos, execute este procedimento uma vez para cada dispositivo enquanto altera os elementos específicos do dispositivo, conforme descrito nos procedimentos.

Procedimentos neste tutorial

- Provisione seu Raspberry Pi para demonstrar (p. 161)
- Configurar aAWS IoTDevice Client para executar o agente de trabalhos (p. 165)

Provisione seu Raspberry Pi para demonstrar

Os procedimentos nesta seção provisionam seu Raspberry Pi emAWS IoTcriandoAWS IoTrecursos e certificados de dispositivo para ele.

Crie e baixe arquivos de certificado de dispositivo para demonstrarAWS IoTempregos

Este procedimento cria os arquivos de certificado do dispositivo para esta demonstração.

Se você estiver preparando mais de um dispositivo, esse procedimento deve ser executado em cada dispositivo.

Para criar e baixar os arquivos de certificado do dispositivo para seu Raspberry Pi:

Na janela do terminal do computador host local que está conectado ao Raspberry Pi, insira esses comandos.

1. Digite o comando a seguir para criar os arquivos de certificado do dispositivo para o dispositivo.

```
aws iot create-keys-and-certificate \
--set-as-active \
--certificate-pem-outfile "~/certs/jobs/device.pem.crt" \
--public-key-outfile "~/certs/jobs/public.pem.key" \
--private-key-outfile "~/certs/jobs/private.pem.key"
```

   O comando retorna uma resposta como a seguinte. Salve o*certificateArn*Para uso posterior.

```
{
"certificateArn": "arn:aws:iot:us-
west-2:57EXAMPLE833:cert/76e7e4edb3e52f52334be2f387a06145b2aa4c7fcd810f3aea2d92abc227d269",
"certificateId": "76e7e4edb3e52f5233EXAMPLE7a06145b2aa4c7fcd810f3aea2d92abc227d269",
"certificatePem": "-----BEGIN CERTIFICATE-----
\nMIIDWTCCAkGgAwIBAgI_SHORTENED_FOR_EXAMPLE_Lgn4jfgtS\n-----END CERTIFICATE-----\n",
"keyPair": {
    "PublicKey": "-----BEGIN PUBLIC KEY-----
\nMIIBIjANBgkqhkiG9w0BA_SHORTENED_FOR_EXAMPLE_ImwIDAQAB\n-----END PUBLIC KEY-----\n",
    "PrivateKey": "-----BEGIN RSA PRIVATE KEY-----
\nMIIEowIBAAKCAQE_SHORTENED_FOR_EXAMPLE_T9RoDiukY\n-----END RSA PRIVATE KEY-----\n"
}
}
```

2. Insira os comandos a seguir para definir as permissões no diretório de certificados e seus arquivos.

```
chmod 700 ~/certs/jobs
chmod 644 ~/certs/jobs/*
chmod 600 ~/certs/jobs/private.pem.key
```

3. Execute este comando para revisar as permissões em seus diretórios e arquivos de certificados.

```
ls -l ~/certs/jobs
```

   A saída do comando deve ser a mesma que você vê aqui, exceto que as datas e horas do arquivo serão diferentes.

```
-rw-r--r-- 1 pi pi 1220 Oct 28 13:02 device.pem.crt
-rw------- 1 pi pi 1675 Oct 28 13:02 private.pem.key
-rw-r--r-- 1 pi pi  451 Oct 28 13:02 public.pem.key
```

Depois de baixar os arquivos de certificado do dispositivo para o Raspberry Pi, você estará pronto para continuarthe section called “Provisione seu Raspberry Pi para demonstrar” (p. 161).

CriarAWS IoTrecursos para demonstrarAWS IoTempregos

Criação doAWS IoTrecursos para este dispositivo.

Se você estiver preparando mais de um dispositivo, esse procedimento deve ser executado para cada dispositivo.

Para provisionar seu dispositivo emAWS IoT:

Na janela do terminal do computador host local que está conectado ao Raspberry Pi:

1. Digite o seguinte comando para obter o endereço do endpoint de dados do dispositivoConta da AWS.

   ```
   aws iot describe-endpoint --endpoint-type IoT:Data-ATS
   ```

   O valor do endpoint não foi alterado desde a última vez que você executou este comando. Executar o comando novamente aqui facilita localizar e colar o valor do endpoint de dados no arquivo de configuração usado neste tutorial.

   Odescribe-endpointO comando retorna uma resposta como a seguinte. Registre o*endpointAddress*Para uso posterior.

   ```
   {
   "endpointAddress": "a3qjEXAMPLEffp-ats.iot.us-west-2.amazonaws.com"
   }
   ```

2. Substituir*UniqueThingName*Com um nome exclusivo para seu dispositivo. Se você quiser executar este tutorial com vários dispositivos, dê a cada dispositivo seu próprio nome. Por exemplo,**TestDevice01**,**TestDevice02**, e assim por diante.

   Digite este comando para criar um novoAWS IoTrecurso para seu Raspberry Pi.

   ```
   aws iot create-thing --thing-name "uniqueThingName"
   ```

   ComoAWS IoTrecurso coisa é umvirtualrepresentação do seu dispositivo na nuvem, podemos criar vários recursos emAWS IoTPara usar para diferentes finalidades. Todos eles podem ser usados pelo mesmo dispositivo de IoT físico para representar diferentes aspectos do dispositivo.

   Esses tutoriais usarão apenas um recurso de cada vez por dispositivo. Dessa forma, nesses tutoriais, eles representam as diferentes demonstrações para que depois de criar oAWS IoTrecursos para uma demonstração, você pode voltar e repetir as demonstrações usando os recursos criados especificamente para cada um.

   Se suas receitasAWS IoTrecurso de coisa foi criado, o comando retorna uma resposta como essa. Registre o*thingArn*valor para uso posteriormente, quando você cria o trabalho a ser executado neste dispositivo.

   ```
   {
   "thingName": "uniqueThingName",
   "thingArn": "arn:aws:iot:us-west-2:57EXAMPLE833:thing/uniqueThingName",
   "thingId": "8ea78707-32c3-4f8a-9232-14bEXAMPLEfd"
   }
   ```

3. Na janela do terminal:

a. Abra um editor de texto, como`nano`.

b. Copie este documento JSON e cole-o no editor de texto aberto.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iot:Connect"
      ],
      "Resource": [
        "arn:aws:iot:us-west-2:57EXAMPLE833:client/uniqueThingName"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "iot:Publish"
      ],
      "Resource": [
        "arn:aws:iot:us-west-2:57EXAMPLE833:topic/test/dc/pubtopic",
        "arn:aws:iot:us-west-2:57EXAMPLE833:topic/$aws/events/job/*",
        "arn:aws:iot:us-west-2:57EXAMPLE833:topic/$aws/events/jobExecution/*",
        "arn:aws:iot:us-west-2:57EXAMPLE833:topic/$aws/things/uniqueThingName/jobs/
*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "iot:Subscribe"
      ],
      "Resource": [
        "arn:aws:iot:us-west-2:57EXAMPLE833:topicfilter/test/dc/subtopic",
        "arn:aws:iot:us-west-2:57EXAMPLE833:topic/$aws/events/jobExecution/*",
        "arn:aws:iot:us-west-2:57EXAMPLE833:topicfilter/$aws/
things/uniqueThingName/jobs/*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "iot:Receive"
      ],
      "Resource": [
        "arn:aws:iot:us-west-2:57EXAMPLE833:topic/test/dc/subtopic",
        "arn:aws:iot:us-west-2:57EXAMPLE833:topic/$aws/things/uniqueThingName/jobs/
*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "iot:DescribeJobExecution",
        "iot:GetPendingJobExecutions",
        "iot:StartNextPendingJobExecution",
        "iot:UpdateJobExecution"
      ],
      "Resource": [
        "arn:aws:iot:us-west-2:57EXAMPLE833:topic/$aws/things/uniqueThingName"
      ]
    }
```

```
  ]
}
```

c. No editor, no`Resource`seção de cada declaração de política, substitua*us-west*Com seuRegião da AWS, um caractere de dois pontos (:) e seus 12 dígitosConta da AWSnúmero.

d. No editor, em cada declaração de política, substitua*UniqueThingName*com o nome da coisa que você deu a essa coisa recurso.

e. Salve o arquivo no editor de texto como**~/policies/jobs_test_thing_policy.json**.

   Se você estiver executando esse procedimento para vários dispositivos, salve o arquivo neste nome de arquivo em cada dispositivo.

4. Substituir*UniqueThingName*com o nome da coisa para o dispositivo e, em seguida, execute este comando para criar umAWS IoTpolítica personalizada para esse dispositivo.

```
aws iot create-policy \
--policy-name "JobTestPolicyForuniqueThingName" \
--policy-document "file://~/policies/jobs_test_thing_policy.json"
```

   Se a política for criada, o comando retornará uma resposta como essa.

```
{
    "policyName": "JobTestPolicyForuniqueThingName",
    "policyArn": "arn:aws:iot:us-west-2:57EXAMPLE833:policy/
JobTestPolicyForuniqueThingName",
    "policyDocument": "{\n\"Version\": \"2012-10-17\",\n\"Statement\": [\n{\n
\"Effect\": \"Allow\",\n\"Action\": [\n\"iot:Connect\"\n],\n\"Resource\": [\n
\"arn:aws:iot:us-west-2:57EXAMPLE833:client/PubSubTestThing\"\n]\n},\n{\n\"Effect\":
 \"Allow\",\n\"Action\": [\n\"iot:Publish\"\n],\n\"Resource\": [\n\"arn:aws:iot:us-
west-2:57EXAMPLE833:topic/test/dc/pubtopic\"\n]\n},\n{\n\"Effect\": \"Allow
\",\n\"Action\": [\n\"iot:Subscribe\"\n],\n\"Resource\": [\n\"arn:aws:iot:us-
west-2:57EXAMPLE833:topicfilter/test/dc/subtopic\"\n]\n},\n{\n\"Effect\": \"Allow
\",\n\"Action\": [\n\"iot:Receive\"\n],\n\"Resource\": [\n\"arn:aws:iot:us-
west-2:57EXAMPLE833:topic/test/dc/*\"\n]\n}\n]\n}\n",
    "policyVersionId": "1"
```

5. Substituir*UniqueThingName*com o nome da coisa para o dispositivo e*certificateArn*com o`certificateArn`valor que você salvou anteriormente nesta seção para este dispositivo e, em seguida, execute este comando para anexar a política ao certificado do dispositivo.

```
aws iot attach-policy \
--policy-name "JobTestPolicyForuniqueThingName" \
--target "certificateArn"
```

   Se houver êxito, o comando não retorna nada.

6. Substituir*UniqueThingName*com o nome da coisa para o dispositivo, substitua*certificateArn*com o`certificateArn`valor que você salvou anteriormente nesta seção e, em seguida, execute este comando para anexar o certificado do dispositivo àAWS IoTrecurso de coisa.

```
aws iot attach-thing-principal \
--thing-name "uniqueThingName" \
--principal "certificateArn"
```

   Se houver êxito, o comando não retorna nada.

Depois de provisionar com sucesso seu Raspberry Pi, você estará pronto para repetir esta seção para outro Raspberry Pi em seu teste ou, se todos os dispositivos tiverem sido provisionados, continuar athe section called “Configurar aAWS IoTDevice Client para executar o agente de trabalhos” (p. 165).

Configurar aAWS IoTDevice Client para executar o agente de trabalhos

Este procedimento cria um arquivo de configuração para oAWS IoTDevice Client para executar o agente de trabalhos:.

Observação: se você estiver preparando mais de um dispositivo, esse procedimento deve ser executado em cada dispositivo.

Para criar o arquivo de configuração para testar oAWS IoTCliente do dispositivo

1. Na janela do terminal do computador host local que está conectado ao Raspberry Pi:

   a. Abra um editor de texto, como`nano`.

   b. Copie este documento JSON e cole-o no editor de texto aberto.

```
{
  "endpoint": "a3qEXAMPLEaffp-ats.iot.us-west-2.amazonaws.com",
  "cert": "~/certs/jobs/device.pem.crt",
  "key": "~/certs/jobs/private.pem.key",
  "root-ca": "~/certs/AmazonRootCA1.pem",
  "thing-name": "uniqueThingName",
  "logging": {
    "enable-sdk-logging": true,
    "level": "DEBUG",
    "type": "STDOUT",
    "file": ""
  },
  "jobs": {
    "enabled": true,
    "handler-directory": ""
  },
  "tunneling": {
    "enabled": false
  },
  "device-defender": {
    "enabled": false,
    "interval": 300
  },
  "fleet-provisioning": {
    "enabled": false,
    "template-name": "",
    "template-parameters": "",
    "csr-file": "",
    "device-key": ""
  },
  "samples": {
    "pub-sub": {
      "enabled": false,
      "publish-topic": "",
      "publish-file": "",
      "subscribe-topic": "",
      "subscribe-file": ""
    }
  },
  "config-shadow": {
    "enabled": false
  },
  "sample-shadow": {
    "enabled": false,
```

```
    "shadow-name": "",
    "shadow-input-file": "",
    "shadow-output-file": ""
  }
}
```

c. Substitua o*Endpoint*valor com o valor do endpoint de dados do dispositivo para seuConta da AWSque você encontrou emthe section called “Provisione seu dispositivo emAWS IoT Core” (p. 141).

d. Substituir*UniqueThingName*com o nome que você usou para este dispositivo.

e. Salve o arquivo no editor de texto como**~/dc-configs/dc-jobs-config.json**.

2. Execute este comando para definir as permissões de arquivo do novo arquivo de configuração.

```
chmod 644 ~/dc-configs/dc-jobs-config.json
```

Você não usará oCliente de teste MQTTpara este teste. Enquanto o dispositivo trocará mensagens MQTT relacionadas a trabalhos comAWS IoT, as mensagens de andamento do trabalho só são trocadas com o dispositivo que executa o trabalho. Como as mensagens de andamento do trabalho são trocadas apenas com o dispositivo que executa o trabalho, você não pode se inscrever nelas de outro dispositivo, como oAWS IoTconsole do .

Depois de salvar o arquivo de configuração, você estará pronto para continuar athe section called “Etapa 2: Criar e executar o trabalho noAWS IoT” (p. 166).

Etapa 2: Criar e executar o trabalho noAWS IoT

Os procedimentos nesta seção criam um documento de trabalho e umAWS IoTrecurso de trabalho. Depois de criar o recurso de trabalho,AWS IoTenvia o documento de trabalho para os destinos de trabalho especificados nos quais um agente de jobs aplica o documento de trabalho ao dispositivo ou cliente.

Procedimentos nesta seção

- Criar e armazenar o documento de trabalho do trabalho (p. 166)
- Executar um trabalhoAWS IoTPara um dispositivo de IoT (p. 167)

Criar e armazenar o documento de trabalho do trabalho

Este procedimento cria um documento de trabalho simples para incluir em umAWS IoTrecurso de trabalho. Este documento de trabalho exibe “Olá, mundo!” no alvo de trabalho.

Para criar e armazenar um documento de trabalho:

1. Selecione o bucket do Amazon S3 no qual você salvará seu documento de trabalho. Se você ainda não tiver um bucket do Amazon S3 para usar para isso, será necessário criar um. Para obter informações sobre como criar buckets do Amazon S3, consulte os tópicos noConceitos básicos do Amazon S3.
2. Criar e salvar o documento de trabalho para este trabalho

   a. No computador host local, abra um editor de texto.

   b. Copie e cole esse texto no editor.

```
{
    "operation": "echo",
    "args": ["Hello world!"]
}
```

c. No computador host local, salve o conteúdo do editor em um arquivo denominado**`hello-world-job.json`**.

d. Confirme se o arquivo foi salvo corretamente. Alguns editores de texto anexam automaticamente`.txt`para o nome do arquivo quando eles salvam um arquivo de texto. Se o editor for anexado`.txt`para o nome do arquivo, corrija o nome do arquivo antes de prosseguir.

3. Substitua o*`path_to_file`*Com o caminho para**`hello-world-job.json`**, se não estiver no diretório atual, substitua*`s3_bucket_name`*com o caminho do bucket do Amazon S3 para o bucket selecionado e, em seguida, execute esse comando para colocar o documento de trabalho no bucket do Amazon S3.

```
aws s3api put-object \
--key hello-world-job.json \
--body path_to_file/hello-world-job.json --bucket s3_bucket_name
```

A URL do documento de trabalho que identifica o documento de trabalho armazenado no Amazon S3 é determinada substituindo o*`s3_bucket_name`*e*`AWS_REGION`*No seguinte URL. Registre o URL resultante para usar posteriormente como o*`job_document_path`*

```
https://s3_bucket_name.s3.AWS_Region.amazonaws.com/hello-world-job.json
```

> **Note**
>
> AWSa segurança impede que você possa abrir esse URL fora do seuConta da AWS, por exemplo, usando um navegador. O URL é usado peloAWS IoTmecanismo de trabalhos, que tem acesso ao arquivo, por padrão. Em um ambiente de produção, você precisará ter certeza de queAWS IoTOs serviços têm permissão para acessar os documentos de trabalho armazenados no Amazon S3.

Depois de salvar o URL do documento de trabalho, continue parathe section called “Executar um trabalhoAWS IoTPara um dispositivo de IoT” (p. 167).

Executar um trabalhoAWS IoTPara um dispositivo de IoT

Os procedimentos nesta seção iniciam oAWS IoTDevice Client em seu Raspberry Pi para executar o agente de trabalhos no dispositivo para aguardar a execução dos trabalhos. Ele também cria um recurso de trabalho noAWS IoT, que enviará o trabalho e será executado em seu dispositivo IoT.

> **Note**
>
> Este procedimento executa uma tarefa em apenas um único dispositivo.

Para iniciar o agente de empregos em seu Raspberry Pi:

1. Na janela do terminal do computador host local que está conectado ao Raspberry Pi, execute este comando para iniciar oAWS IoTCliente do dispositivo.

```
cd ~/aws-iot-device-client/build
./aws-iot-device-client --config-file ~/dc-configs/dc-jobs-config.json
```

2. Na janela do terminal, confirme que oAWS IoTDevice Client e exibe essas mensagens

```
2021-11-15T18:45:56.708Z [INFO]  {Main.cpp}: Jobs is enabled
                        .
                        .
                        .
2021-11-15T18:45:56.708Z [INFO]  {Main.cpp}: Client base has been notified that Jobs
 has started
```

```
2021-11-15T18:45:56.708Z [INFO]  {JobsFeature.cpp}: Running Jobs!
2021-11-15T18:45:56.708Z [DEBUG] {JobsFeature.cpp}: Attempting to subscribe to
 startNextPendingJobExecution accepted and rejected
2021-11-15T18:45:56.708Z [DEBUG] {JobsFeature.cpp}: Attempting to subscribe to
 nextJobChanged events
2021-11-15T18:45:56.708Z [DEBUG] {JobsFeature.cpp}: Attempting to subscribe to
 updateJobExecutionStatusAccepted for jobId +
2021-11-15T18:45:56.738Z [DEBUG] {JobsFeature.cpp}: Ack received for
 SubscribeToUpdateJobExecutionAccepted with code {0}
2021-11-15T18:45:56.739Z [DEBUG] {JobsFeature.cpp}: Attempting to subscribe to
 updateJobExecutionStatusRejected for jobId +
2021-11-15T18:45:56.753Z [DEBUG] {JobsFeature.cpp}: Ack received for
 SubscribeToNextJobChanged with code {0}
2021-11-15T18:45:56.760Z [DEBUG] {JobsFeature.cpp}: Ack received for
 SubscribeToStartNextJobRejected with code {0}
2021-11-15T18:45:56.776Z [DEBUG] {JobsFeature.cpp}: Ack received for
 SubscribeToStartNextJobAccepted with code {0}
2021-11-15T18:45:56.776Z [DEBUG] {JobsFeature.cpp}: Ack received for
 SubscribeToUpdateJobExecutionRejected with code {0}
2021-11-15T18:45:56.777Z [DEBUG] {JobsFeature.cpp}: Publishing
 startNextPendingJobExecutionRequest
2021-11-15T18:45:56.785Z [DEBUG] {JobsFeature.cpp}: Ack received for
 StartNextPendingJobPub with code {0}
2021-11-15T18:45:56.785Z [INFO]  {JobsFeature.cpp}: No pending jobs are scheduled,
 waiting for the next incoming job
```

3. Na janela do terminal, depois de ver esta mensagem, continue para o próximo procedimento e crie o recurso de trabalho. Observe que talvez não seja a última entrada na lista.

```
2021-11-15T18:45:56.785Z [INFO]  {JobsFeature.cpp}: No pending jobs are scheduled,
 waiting for the next incoming job
```

Para criar umAWS IoTrecurso de trabalho

1. No computador host local:

   a. Substituir*job_document_url*Com o URL do documento de trabalhothe section called “Criar e armazenar o documento de trabalho do trabalho” (p. 166).

   b. Substituir*thing_arn*com o ARN do recurso coisa que você criou para o seu dispositivo e, em seguida, execute esse comando.

```
aws iot create-job \
--job-id hello-world-job-1 \
--document-source "job_document_url" \
--targets "thing_arn" \
--target-selection SNAPSHOT
```

   Se houver êxito, o comando retornará um resultado como este.

```
{
  "jobArn": "arn:aws:iot:us-west-2:57EXAMPLE833:job/hello-world-job-1",
  "jobId": "hello-world-job-1"
}
```

2. Na janela do terminal, você deve ver a saída doAWS IoTDevice Client assim.

```
2021-11-15T18:02:26.688Z [INFO]  {JobsFeature.cpp}: No pending jobs are scheduled,
 waiting for the next incoming job
2021-11-15T18:10:24.890Z [DEBUG] {JobsFeature.cpp}: Job ids differ
```

```
2021-11-15T18:10:24.890Z [INFO]  {JobsFeature.cpp}: Executing job: hello-world-job-1
2021-11-15T18:10:24.890Z [DEBUG] {JobsFeature.cpp}: Attempting to update job execution
 status!
2021-11-15T18:10:24.890Z [DEBUG] {JobsFeature.cpp}: Not including stdout with the
 status details
2021-11-15T18:10:24.890Z [DEBUG] {JobsFeature.cpp}: Not including stderr with the
 status details
2021-11-15T18:10:24.890Z [DEBUG] {JobsFeature.cpp}: Assuming executable is in PATH
2021-11-15T18:10:24.890Z [INFO]  {JobsFeature.cpp}: About to execute: echo Hello world!
2021-11-15T18:10:24.890Z [DEBUG] {Retry.cpp}: Retryable function starting, it will
 retry until success
2021-11-15T18:10:24.890Z [DEBUG] {JobsFeature.cpp}: Created EphermalPromise for
 ClientToken 3TEWba9Xj6 in the updateJobExecution promises map
2021-11-15T18:10:24.890Z [DEBUG] {JobEngine.cpp}: Child process now running
2021-11-15T18:10:24.890Z [DEBUG] {JobEngine.cpp}: Child process about to call execvp
2021-11-15T18:10:24.890Z [DEBUG] {JobEngine.cpp}: Parent process now running, child PID
 is 16737
2021-11-15T18:10:24.891Z [DEBUG] {16737}: Hello world!
2021-11-15T18:10:24.891Z [DEBUG] {JobEngine.cpp}: JobEngine finished waiting for child
 process, returning 0
2021-11-15T18:10:24.891Z [INFO]  {JobsFeature.cpp}: Job exited with status: 0
2021-11-15T18:10:24.891Z [INFO]  {JobsFeature.cpp}: Job executed successfully!
2021-11-15T18:10:24.891Z [DEBUG] {JobsFeature.cpp}: Attempting to update job execution
 status!
2021-11-15T18:10:24.891Z [DEBUG] {JobsFeature.cpp}: Not including stdout with the
 status details
2021-11-15T18:10:24.891Z [DEBUG] {JobsFeature.cpp}: Not including stderr with the
 status details
2021-11-15T18:10:24.892Z [DEBUG] {Retry.cpp}: Retryable function starting, it will
 retry until success
2021-11-15T18:10:24.892Z [DEBUG] {JobsFeature.cpp}: Created EphermalPromise for
 ClientToken GmQ0HTzWGg in the updateJobExecution promises map
2021-11-15T18:10:24.905Z [DEBUG] {JobsFeature.cpp}: Ack received for
 PublishUpdateJobExecutionStatus with code {0}
2021-11-15T18:10:24.905Z [DEBUG] {JobsFeature.cpp}: Removing ClientToken 3TEWba9Xj6
 from the updateJobExecution promises map
2021-11-15T18:10:24.905Z [DEBUG] {JobsFeature.cpp}: Success response after
 UpdateJobExecution for job hello-world-job-1
2021-11-15T18:10:24.917Z [DEBUG] {JobsFeature.cpp}: Ack received for
 PublishUpdateJobExecutionStatus with code {0}
2021-11-15T18:10:24.918Z [DEBUG] {JobsFeature.cpp}: Removing ClientToken GmQ0HTzWGg
 from the updateJobExecution promises map
2021-11-15T18:10:24.918Z [DEBUG] {JobsFeature.cpp}: Success response after
 UpdateJobExecution for job hello-world-job-1
2021-11-15T18:10:25.861Z [INFO]  {JobsFeature.cpp}: No pending jobs are scheduled,
 waiting for the next incoming job
```

3. Enquanto oAWS IoTO Device Client está em execução e aguardando um trabalho, você pode enviar outro trabalho alterando o`job-id`Para executar novamente ocreate-jobNa Etapa 1.

Quando você terminar de executar trabalhos, na janela do terminal, insira^C(Control-C) para parar oAWS IoTCliente do dispositivo.

Tutorial: Limpeza após a execução doAWS IoTTutoriais do dispositivo cliente

Os procedimentos neste tutorial orientam você na remoção dos arquivos e recursos que você criou enquanto conclui os tutoriais neste caminho de aprendizado.

Procedimentos neste tutorial

- Etapa 1: Limpando seus dispositivos depois de criar demonstrações com oAWS IoTCliente do dispositivo (p. 170)
- Etapa 2: Limpando oConta da AWSdepois de criar demonstrações com oAWS IoTCliente do dispositivo (p. 171)

Etapa 1: Limpando seus dispositivos depois de criar demonstrações com oAWS IoTCliente do dispositivo

Este tutorial descreve duas opções de como limpar o cartão microSD depois de criar as demonstrações neste caminho de aprendizado. Escolha a opção que fornece o nível de segurança que você precisa.

Observe que a limpeza do cartão microSD do dispositivo não remove nenhumAWS IoTRecursos que você criou. Para limpar oAWS IoTrecursos depois de limpar o cartão microSD do dispositivo, você deve revisar o tutorial emthe section called “Limpeza após a criação de demonstrações com oAWS IoTCliente do dispositivo” (p. 171).

Opção 1: Limpando reescrevendo o cartão microSD

A maneira mais fácil e completa de limpar o cartão microSD depois de concluir os tutoriais neste caminho de aprendizado é substituir o cartão microSD por um arquivo de imagem salvo que você criou durante a preparação do dispositivo pela primeira vez.

Este procedimento usa o computador host local para gravar uma imagem de cartão microSD salva em um cartão microSD.

> **Note**
>
> Se o dispositivo não usar um meio de armazenamento removível para o sistema operacional, consulte o procedimento desse dispositivo.

Para gravar uma nova imagem no cartão microSD

1. No computador host local, localize a imagem do cartão microSD salva que você deseja gravar no cartão microSD.
2. Insira seu cartão microSD no computador host local.
3. Usando uma ferramenta de imagem de cartão SD, escreva o arquivo de imagem selecionado no cartão microSD.
4. Depois de gravar a imagem do Raspberry Pi OS no cartão microSD, ejete o cartão microSD e remova-o com segurança do computador host local.

Seu cartão microSD está pronto para uso.

Opção 2: Limpando excluindo diretórios de usuários

Para limpar o cartão microSD depois de concluir os tutoriais sem reescrever a imagem do cartão microSD, você pode excluir os diretórios do usuário individualmente. Isso não é tão completo quanto reescrever o cartão microSD de uma imagem salva porque ele não remove nenhum arquivo do sistema que possa ter sido instalado.

Se a remoção dos diretórios do usuário for suficientemente minuciosa para suas necessidades, você poderá seguir este procedimento.

Para excluir os diretórios de usuário desse caminho de aprendizado do seu dispositivo

1. Execute esses comandos para excluir os diretórios de usuário, subdiretórios e todos os arquivos que foram criados nesse caminho de aprendizado, na janela do terminal conectada ao seu dispositivo.

Note

Depois de excluir esses diretórios e arquivos, você não poderá executar as demonstrações sem concluir os tutoriais novamente.

```
rm -Rf ~/dc-configs
rm -Rf ~/policies
rm -Rf ~/messages
rm -Rf ~/certs
rm -Rf ~/.aws-iot-device-client
```

2. Execute esses comandos para excluir os diretórios e arquivos de origem do aplicativo, na janela do terminal conectada ao seu dispositivo.

   Note

   Esses comandos não desinstalam nenhum programa. Eles só removem os arquivos de origem usados para compilá-los e instalá-los. Depois de excluir esses arquivos, oAWS CLIO e aAWS IoTO dispositivo cliente pode não funcionar.

```
rm -Rf ~/aws-cli
rm -Rf ~/aws
rm -Rf ~/aws-iot-device-client
```

Etapa 2: Limpando oConta da AWSdepois de criar demonstrações com oAWS IoTCliente do dispositivo

Esses procedimentos ajudam a identificar e remover oAWSrecursos que você criou ao concluir os tutoriais neste caminho de aprendizado.

LimparAWS IoTrecursos

Este procedimento ajuda você a identificar e remover oAWS IoTrecursos que você criou ao concluir os tutoriais neste caminho de aprendizado.

AWS IoTrecursos criados neste caminho de aprendizado

Tutorial	Recurso de coisas	Recurso de políticas
the section called “Instalar e configurar oAWS IoTCliente do dispositivo” (p. 137)	DevClitestThing	DevClitestThingPolicy
the section called “Demonstrar comunicação de mensagens MQTT com oAWS IoTCliente do dispositivo” (p. 146)	PubSubtestthing	PubSubTestThingPolicy
the section called “Demonstrar ações remotas (trabalhos) com oAWS IoTCliente do dispositivo” (p. 160)	Definido pelo usuário(pode haver mais de um)	Definido pelo usuário(pode haver mais de um)

Como excluir oAWS IoTrecursos, siga este procedimento para cada recurso que você criou

1. Substituir*thing_name*com o nome do recurso que você deseja excluir e, em seguida, execute este comando para listar os certificados anexados ao recurso coisa, a partir do computador host local.

```
aws iot list-thing-principals --thing-name thing_name
```

Esse comando retorna uma resposta como esta que lista os certificados anexados a*thing_name*. Na maioria dos casos, haverá apenas um certificado na lista.

```
{
    "principals": [
        "arn:aws:iot:us-
west-2:57EXAMPLE833:cert/23853eea3cf0edc7f8a69c74abeafa27b2b52823cab5b3e156295e94b26ae8ac"
    ]
}
```

2. Para cada certificado listado pelo comando anterior:

   a. Substituir*certificate_ID*Com o ID do certificado do comando anterior. O ID do certificado são os caracteres alfanuméricos que se seguem`cert/`no ARN retornado pelo comando anterior. Em seguida, execute este comando para desativar o certificado.

   ```
   aws iot update-certificate --new-status INACTIVE --certificate-id certificate_ID
   ```

   Se houver êxito, o comando não retornará nada.

   b. Substituir*certificate_ARN*com o certificado ARN da lista de certificados retornados anteriormente e, em seguida, execute este comando para listar as políticas anexadas a este certificado.

   ```
   aws iot list-attached-policies --target certificate_ARN
   ```

   Esse comando retorna uma resposta como esta que lista as políticas anexadas ao certificado. Na maioria dos casos, haverá apenas uma política na lista.

   ```
   {
       "policies": [
           {
               "policyName": "DevCliTestThingPolicy",
               "policyArn": "arn:aws:iot:us-west-2:57EXAMPLE833:policy/
   DevCliTestThingPolicy"
           }
       ]
   }
   ```

   c. Para cada política anexada ao certificado:

      i. Substituir*policy_name*com o`policyName`valor do comando anterior, substitua*certificate_ARN*com o ARN do certificado e, em seguida, execute este comando para desanexar a política do certificado.

      ```
      aws iot detach-policy --policy-name policy_name --target certificate_ARN
      ```

      Se houver êxito, o comando não retornará nada.

      ii. Substituir*policy_name*com o`policyName`value e, em seguida, execute este comando para ver se a política está anexada a mais certificados.

```
aws iot list-targets-for-policy --policy-name policy_name
```

Se o comando retornar uma lista vazia como esta, a política não será anexada a nenhum certificado e você continuará listando as versões da política. Se ainda houver certificados anexados à política, continue com adetach-thing-principalpasso.

```
{
    "targets": []
}
```

iii. Substituir*policy_name*com `opolicyName`value e, em seguida, execute este comando para verificar as versões da política. Para excluir a política, ela deve ter apenas uma versão.

```
aws iot list-policy-versions --policy-name policy_name
```

Se a política tiver apenas uma versão, como este exemplo, você pode pular para adelete-policypasso e exclua a política agora.

```
{
    "policyVersions": [
        {
            "versionId": "1",
            "isDefaultVersion": true,
            "createDate": "2021-11-18T01:02:46.778000+00:00"
        }
    ]
}
```

Se a política tiver mais de uma versão, como este exemplo, as versões da política com um`isDefaultVersion`valor de`false`deve ser excluído antes que a política possa ser excluída.

```
{
    "policyVersions": [
        {
            "versionId": "2",
            "isDefaultVersion": true,
            "createDate": "2021-11-18T01:52:04.423000+00:00"
        },
        {
            "versionId": "1",
            "isDefaultVersion": false,
            "createDate": "2021-11-18T01:30:18.083000+00:00"
        }
    ]
}
```

Se você precisar excluir uma versão da política, substitua*policy_name*com `opolicyName`valor, substitua*version_ID*com `oversionId`valor do comando anterior e, em seguida, execute este comando para excluir uma versão de política.

```
aws iot delete-policy-version --policy-name policy_name --policy-version-
id version_ID
```

Se houver êxito, o comando não retornará nada.

Depois de excluir uma versão de política, repita essa etapa até que a política tenha apenas uma versão da política.

iv. Substituir*policy_name*com o`policyName`value e, em seguida, execute este comando para excluir a política.

```
aws iot delete-policy --policy-name policy_name
```

d. Substituir*thing_name*com o nome da coisa, substitua*certificate_ARN*com o ARN do certificado e, em seguida, execute este comando para desanexar o certificado do recurso coisa.

```
aws iot detach-thing-principal --thing-name thing_name --principal certificate_ARN
```

Se houver êxito, o comando não retornará nada.

e. Substituir*certificate_ID*Com o ID do certificado do comando anterior. O ID do certificado são os caracteres alfanuméricos que se seguem`cert/`no ARN retornado pelo comando anterior. Em seguida, execute o comando para excluir o recurso do certificado.

```
aws iot delete-certificate --certificate-id certificate_ID
```

Se houver êxito, o comando não retornará nada.

3. Substituir*thing_name*com o nome da coisa e, em seguida, execute esse comando para excluir a coisa.

```
aws iot delete-thing --thing-name thing_name
```

Se houver êxito, o comando não retornará nada.

LimparAWSrecursos

Este procedimento ajuda você a identificar e remover outrosAWSrecursos que você criou ao concluir os tutoriais neste caminho de aprendizado.

Other (Outros)AWSrecursos criados neste caminho de aprendizado

Tutorial	Tipo de recurso	Nome ou ID do recurso
the section called “Demonstrar ações remotas (trabalhos) com oAWS IoTCliente do dispositivo” (p. 160)	Objeto Amazon S3	hello-world-job.json
the section called “Demonstrar ações remotas (trabalhos) com oAWS IoTCliente do dispositivo” (p. 160)	AWS IoTrecursos de trabalho	Definido pelo usuário

Como excluir oAWSrecursos criados neste caminho de aprendizado

1. Para excluir os trabalhos criados neste caminho de aprendizado

   a. Execute este comando para listar os trabalhos em seuConta da AWS.

```
aws iot list-jobs
```

O comando retorna uma lista doAWS IoTjobs noConta da AWSeRegião da AWSSe parece com isso.

```
{
    "jobs": [
        {
            "jobArn": "arn:aws:iot:us-west-2:57EXAMPLE833:job/hello-world-job-2",
            "jobId": "hello-world-job-2",
            "targetSelection": "SNAPSHOT",
            "status": "COMPLETED",
            "createdAt": "2021-11-16T23:40:36.825000+00:00",
            "lastUpdatedAt": "2021-11-16T23:40:41.375000+00:00",
            "completedAt": "2021-11-16T23:40:41.375000+00:00"
        },
        {
            "jobArn": "arn:aws:iot:us-west-2:57EXAMPLE833:job/hello-world-job-1",
            "jobId": "hello-world-job-1",
            "targetSelection": "SNAPSHOT",
            "status": "COMPLETED",
            "createdAt": "2021-11-16T23:35:26.381000+00:00",
            "lastUpdatedAt": "2021-11-16T23:35:29.239000+00:00",
            "completedAt": "2021-11-16T23:35:29.239000+00:00"
        }
    ]
}
```

b. Para cada trabalho que você reconhece na lista como um trabalho criado nesse caminho de aprendizado, substitua*jobId*com `ojobId`valor do trabalho a ser excluído e, em seguida, execute este comando para excluir umAWS IoTtrabalho.

```
aws iot delete-job --job-id jobId
```

Se o comando for bem-sucedido, não retornará nada.

2. Para excluir os documentos de trabalho que você armazenou em um bucket do Amazon S3 neste caminho de aprendizado.

   a. Substituir*bucket*com o nome do bucket que você usou e, em seguida, execute este comando para listar os objetos no bucket do Amazon S3 que você usou.

```
aws s3api list-objects --bucket bucket
```

O comando retorna uma lista dos objetos do Amazon S3 no bucket que se parece com isso.

```
{
    "Contents": [
        {
            "Key": "hello-world-job.json",
            "LastModified": "2021-11-18T03:02:12+00:00",
            "ETag": "\"868c8bc3f56b5787964764d4b18ed5ef\"",
            "Size": 54,
            "StorageClass": "STANDARD",
            "Owner": {
                "DisplayName": "EXAMPLE",
                "ID":
 "e9e3d6ec1EXAMPLEf5bfb5e6bd0a2b6ed03884d1ed392a82ad011c144736a4ee"
            }
```

```
        },
        {
            "Key": "iot_job_firmware_update.json",
            "LastModified": "2021-04-13T21:57:07+00:00",
            "ETag": "\"7c68c591949391791ecf625253658c61\"",
            "Size": 66,
            "StorageClass": "STANDARD",
            "Owner": {
                "DisplayName": "EXAMPLE",
                "ID":
 "e9e3d6ec1EXAMPLEf5bfb5e6bd0a2b6ed03884d1ed392a82ad011c144736a4ee"
            }
        },
        {
            "Key": "order66.json",
            "LastModified": "2021-04-13T21:57:07+00:00",
            "ETag": "\"bca60d5380b88e1a70cc27d321caba72\"",
            "Size": 29,
            "StorageClass": "STANDARD",
            "Owner": {
                "DisplayName": "EXAMPLE",
                "ID":
 "e9e3d6ec1EXAMPLEf5bfb5e6bd0a2b6ed03884d1ed392a82ad011c144736a4ee"
            }
        }
    ]
}
```

b. Para cada objeto que você reconhece da lista como um objeto criado nesse caminho de aprendizado, substitua*bucket*com o nome do bucket e*key*Com o valor da chave do objeto a ser excluído e execute o comando para excluir um objeto do Amazon S3.

```
aws s3api delete-object --bucket bucket --key key
```

Se o comando for bem-sucedido, não retornará nada.

Depois de excluir todos osAWSrecursos e objetos que você criou ao concluir esse caminho de aprendizado, você pode começar de novo e repetir os tutoriais.

Construindo soluções com oAWS IoTSDKs de dispositivo da

Os tutoriais desta seção ajudam você a orientar as etapas para desenvolver uma solução de IoT que pode ser implantada em um ambiente de produção usandoAWS IoT.

Esses tutoriais podem levar mais tempo para serem concluídos do que os da seção sobrethe section called “Criando demonstrações com oAWS IoTCliente de dispositivo” (p. 125)porque eles usam oAWS IoTSDKs do dispositivo e explique os conceitos que estão sendo aplicados com mais detalhes para ajudá-lo a criar soluções seguras e confiáveis.

Comece a criar soluções com oAWS IoTSDKs de dispositivo da

Esses tutoriais orientam você através de diferentesAWS IoTCenários. Quando apropriado, os tutoriais usam aAWS IoTOs SDKs de dispositivo.

Tópicos

- Tutorial: Conectar um dispositivo àAWS IoT Coreusando oAWS IoTSDK de dispositivo da (p. 177)
- CriarAWS IoTregras para rotear dados do dispositivo para outros serviços (p. 193)
- Mantendo o estado do dispositivo enquanto o dispositivo está offline com Sombras de dispositivos (p. 223)
- Tutorial: Criação de um autorizador personalizado paraAWS IoT Core (p. 243)
- Tutorial: Monitoramento da umidade do soloAWS IoTe Raspberry Pi (p. 255)

Tutorial: Conectar um dispositivo àAWS IoT Coreusando oAWS IoTSDK de dispositivo da

Este tutorial demonstra como conectar um dispositivo aoAWS IoT Corepara que ele possa enviar e receber dados de e paraAWS IoT. Depois de concluir este tutorial, seu dispositivo será configurado para se conectar aoAWS IoT Coree você entenderá como os dispositivos se comunicam comAWS IoT.

Neste tutorial, você vai:

1. the section called “Prepare seu dispositivo paraAWS IoT” (p. 177)
2. the section called “Revise o protocolo MQTT” (p. 178)
3. the section called “Revise o aplicativo de exemplo do SDK do dispositivo pubsub.py” (p. 178)
4. the section called “Connect o dispositivo e comunique comAWS IoT Core” (p. 184)
5. the section called “Analise os resultados” (p. 190)

Este tutorial leva cerca de uma hora para ser concluído.

Antes de começar este tutorial, certifique-se de que você tenha:

- CompletedConceitos básicos do AWS IoT Core (p. 18)

  Na seção desse tutorial, onde você devethe section called “Configurar o dispositivo” (p. 43), selecione othe section called “Connect um Raspberry Pi ou outro dispositivo” (p. 57)opção para o dispositivo e use as opções de idioma Python para configurar seu dispositivo.

  Mantenha aberta a janela do terminal que você usa nesse tutorial porque você também a usará neste tutorial.
- Um dispositivo que pode executar oAWS IoTSDK de dispositivo v2 para Python.

  Este tutorial mostra como conectar um dispositivo aoAWS IoT Coreusando exemplos de código Python, que exigem um dispositivo relativamente poderoso.

  Se você estiver trabalhando com dispositivos restritos de recursos, esses exemplos de código podem não funcionar neles. Nesse caso, talvez tenha mais sucesso aothe section called “Usar a AWS IoT Device SDK para C incorporado” (p. 190)para "Hello, World!".

Prepare seu dispositivo paraAWS IoT

DentroConceitos básicos do AWS IoT Core (p. 18), você preparou seu dispositivo eAWSconta para que eles pudessem se comunicar. Esta seção analisa os aspectos dessa preparação que se aplicam a qualquer conexão de dispositivo comAWS IoT Core.

Para que um dispositivo se conecte aAWS IoT Core:

1. Você deve ter umConta da AWS.

   O procedimento emConfigurar seuConta da AWS (p. 19)Descreve como criar umConta da AWSSe você ainda não tiver um.
2. Nessa conta, você deve ter o seguinteAWS IoTrecursosdefinido para o dispositivo em seuConta da AWSe Região.

   O procedimento emCriarAWS IoTrecursos (p. 39)descreve como criar esses recursos para o dispositivo em seuConta da AWSe Região.
   - UMACertificado de dispositivoRegistrado comAWS IoTe ativado para autenticar o dispositivo.

     O certificado geralmente é criado com e anexado a umAWS IoTobjeto da coisa. Embora um objeto coisa não seja necessário para que um dispositivo se conecte aoAWS IoT, ele torna adicionalAWS IoTrecursos disponíveis para o dispositivo.
   - UMAregraanexado ao certificado do dispositivo que o autoriza a se conectar aoAWS IoT Coree execute todas as ações que você deseja.
3. UmaConexão com a Internetque podem acessar seuConta da AWSendpoints do dispositivo.

   Os endpoints do dispositivo são descritos emAWS IoTdados do dispositivo e endpoints de serviço (p. 80)e pode ser visto noPágina de configurações doAWS IoTconsole.
4. Software de comunicaçãocomo oAWS IoTOs SDKs de dispositivo fornecem. Este tutorial usa oAWS IoTSDK de dispositivo v2 para Python.

Revise o protocolo MQTT

Antes de falarmos sobre o aplicativo de amostra, ajuda a entender o protocolo MQTT. O protocolo MQTT oferece algumas vantagens em relação a outros protocolos de comunicação de rede, como HTTP, o que o torna uma escolha popular para dispositivos IoT. Esta seção analisa os principais aspectos do MQTT que se aplicam a este tutorial. Para obter informações sobre como o MQTT se compara ao ao HTTP, consulteEscolhendo um protocolo para a comunicação do seu dispositivo (p. 85).

O MQTT usa um modelo de comunicação de publicação/assinatura

O protocolo MQTT usa um modelo de comunicação de publicação/assinatura com seu host. Esse modelo difere do modelo de solicitação/resposta do que o HTTP usa. Com o MQTT, os dispositivos estabelecem uma sessão com o host identificado por um ID de cliente exclusivo. Para enviar dados, os dispositivos publicam mensagens identificadas por tópicos para um corretor de mensagens no host. Para receber mensagens do corretor de mensagens, os dispositivos se inscrevem em tópicos enviando filtros de tópicos em solicitações de assinatura para o corretor de mensagens.

MQTT suporta sessões persistentes

O corretor de mensagens recebe mensagens de dispositivos e publica mensagens em dispositivos que se inscreveram nelas. comsessões persistentes (p. 88)—sessões que permanecem ativas mesmo quando o dispositivo iniciador está desconectado — os dispositivos podem recuperar mensagens que foram publicadas enquanto estavam desconectadas. No lado do dispositivo, o MQTT suporta níveis de Qualidade de Serviço (QoS (p. 88)) que garantem que o host receba mensagens enviadas pelo dispositivo.

Revise o aplicativo de exemplo do SDK do dispositivo pubsub.py

Esta seção analisa `opubsub.py`Exemplo de aplicativo doAWS IoTSDK de dispositivo v2 para Pythonusado neste tutorial. Aqui, analisaremos como ele se conecta aoAWS IoT Corepara publicar e assinar mensagens MQTT. A próxima seção apresenta alguns exercícios para ajudá-lo a explorar como um dispositivo se conecta e se comunica comAWS IoT Core.

Opubsub.pyaplicativo de amostra demonstra esses aspectos de uma conexão MQTT comAWS IoT Core:

- Protocolos de (p. 179)
- Sessões persistentes (p. 181)
- Qualidade de serviço (p. 182)
- Publicar mensagens (p. 182)
- Assinatura de (p. 183)
- Desconexão e reconexão do dispositivo (p. 184)

Protocolos de

O`pubsub.py`sample demonstra uma conexão MQTT usando os protocolos MQTT e MQTT sobre WSS. OAWSTempo de execução comum (AWSCRT)biblioteca fornece suporte ao protocolo de comunicação de baixo nível e está incluída noAWS IoTSDK de dispositivo v2 para Python.

MQTT

O`pubsub.py`Chamadas de exemplo`mtls_from_path`(mostrado aqui) no`mqtt_connection_builder`Para estabelecer uma conexão comAWS IoT Coreusando o protocolo MQTT.`mtls_from_path`usa certificados X.509 e TLS v1.2 para autenticar o dispositivo. OAWSA biblioteca CRT lida com os detalhes de nível inferior dessa conexão.

```
mqtt_connection = mqtt_connection_builder.mtls_from_path(
    endpoint=args.endpoint,
    cert_filepath=args.cert,
    pri_key_filepath=args.key,
    ca_filepath=args.ca_file,
    client_bootstrap=client_bootstrap,
    on_connection_interrupted=on_connection_interrupted,
    on_connection_resumed=on_connection_resumed,
    client_id=args.client_id,
    clean_session=False,
    keep_alive_secs=6
)
```

`endpoint`

SuasConta da AWSendpoint do dispositivo IoT da

No aplicativo de amostra, esse valor é transmitido da linha de comando.

`cert_filepath`

O caminho para o arquivo de certificado do dispositivo

No aplicativo de amostra, esse valor é transmitido da linha de comando.

`pri_key_filepath`

O caminho para o arquivo de chave privada do dispositivo que foi criado com seu arquivo de certificado

No aplicativo de amostra, esse valor é transmitido da linha de comando.

`ca_filepath`

O caminho para o arquivo CA raiz da. Obrigatório somente se o servidor MQTT usar um certificado que ainda não esteja em seu repositório de confiança.

No aplicativo de amostra, esse valor é transmitido da linha de comando.

`client_bootstrap`

O objeto de tempo de execução comum que lida com as atividades de comunicação de

No aplicativo de amostra, esse objeto é instanciado antes da chamada para`mqtt_connection_builder.mtls_from_path`.

`on_connection_interrupted`, `on_connection_resumed`

As funções de retorno de chamada a serem chamadas quando a conexão do dispositivo é interrompida e retomada

`client_id`

O ID que identifica exclusivamente este dispositivo noRegião da AWS

No aplicativo de amostra, esse valor é transmitido da linha de comando.

`clean_session`

Se deseja iniciar uma nova sessão persistente ou, se houver uma, reconecte-se a uma existente

`keep_alive_secs`

O valor de manter vivo, em segundos, para enviar o`CONNECT`solicitação. Um ping será enviado automaticamente nesse intervalo. Se o servidor não receber um ping após 1,5 vezes esse valor, ele pressupõe que a conexão seja perdida.

MQTT por WSS

O`pubsub.py`Chamadas de exemplo`websockets_with_default_aws_signing`(mostrado aqui) no`mqtt_connection_builder`Para estabelecer uma conexão comAWS IoT Coreusar o protocolo MQTT sobre o WSS.`websockets_with_default_aws_signing`cria uma conexão MQTT via WSS usandoSignature V4para autenticar o dispositivo.

```
mqtt_connection = mqtt_connection_builder.websockets_with_default_aws_signing(
    endpoint=args.endpoint,
    client_bootstrap=client_bootstrap,
    region=args.signing_region,
    credentials_provider=credentials_provider,
    websocket_proxy_options=proxy_options,
    ca_filepath=args.ca_file,
    on_connection_interrupted=on_connection_interrupted,
    on_connection_resumed=on_connection_resumed,
    client_id=args.client_id,
    clean_session=False,
    keep_alive_secs=6
)
```

`endpoint`

SuasConta da AWSendpoint do dispositivo IoT da

No aplicativo de amostra, esse valor é transmitido da linha de comando.

`client_bootstrap`

O objeto de tempo de execução comum que lida com as atividades de comunicação de

No aplicativo de amostra, esse objeto é instanciado antes da chamada para`mqtt_connection_builder.websockets_with_default_aws_signing`.

`region`

OAWSassinatura Região usada pela autenticação Signature V4. Dentro`pubsub.py`, ele passa o parâmetro inserido na linha de comando.

No aplicativo de amostra, esse valor é transmitido da linha de comando.

`credentials_provider`

OAWScredenciais fornecidas para uso para autenticação

No aplicativo de amostra, esse objeto é instanciado antes da chamada para`mqtt_connection_builder.websockets_with_default_aws_signing`.

`websocket_proxy_options`

Opções de proxy HTTP, se estiver usando um host proxy

No aplicativo de amostra, esse valor é inicializado antes da chamada para`mqtt_connection_builder.websockets_with_default_aws_signing`.

`ca_filepath`

O caminho para o arquivo CA raiz da. Obrigatório somente se o servidor MQTT usar um certificado que ainda não esteja em seu repositório de confiança.

No aplicativo de amostra, esse valor é transmitido da linha de comando.

`on_connection_interrupted`, `on_connection_resumed`

As funções de retorno de chamada a serem chamadas quando a conexão do dispositivo é interrompida e retomada

`client_id`

O ID que identifica exclusivamente este dispositivo noRegião da AWS.

No aplicativo de amostra, esse valor é transmitido da linha de comando.

`clean_session`

Se deseja iniciar uma nova sessão persistente ou, se houver uma, reconecte-se a uma existente

`keep_alive_secs`

O valor de manter vivo, em segundos, para enviar o`CONNECT`solicitação. Um ping será enviado automaticamente nesse intervalo. Se o servidor não receber um ping após 1,5 vezes esse valor, ele pressupõe que a conexão seja perdida.

HTTPS

E HTTPS?AWS IoT Coresuporta dispositivos que publicam solicitações HTTPS. Do ponto de vista da programação, os dispositivos enviam solicitações HTTPS paraAWS IoT Coreassim como qualquer outro aplicativo. Para obter um exemplo de programa Python que envia uma mensagem HTTP de um dispositivo, consulte aExemplo de código HTTPS (p. 97)Usando o Python`requests`biblioteca. Este exemplo envia uma mensagem paraAWS IoT Coreusando HTTPS de tal forma queAWS IoT CoreInterprete-o como uma mensagem MQTT.

EnquantoAWS IoT Coresuporta solicitações HTTPS de dispositivos, certifique-se de revisar as informações sobreEscolhendo um protocolo para a comunicação do seu dispositivo (p. 85)para que você possa tomar uma decisão informada sobre qual protocolo usar para as comunicações do seu dispositivo.

Sessões persistentes

No aplicativo de amostra, definindo o`clean_session`Parâmetro para`False`indica que a conexão deve ser persistente. Na prática, isso significa que a conexão aberta por essa chamada se reconecta a uma sessão persistente existente, se houver. Caso contrário, ele cria e se conecta a uma nova sessão persistente.

Com uma sessão persistente, as mensagens enviadas para o dispositivo são armazenadas pelo agente de mensagens enquanto o dispositivo não está conectado. Quando um dispositivo se reconecta a uma sessão persistente, o agente de mensagens envia ao dispositivo todas as mensagens armazenadas nas quais ele se inscreveu.

Sem uma sessão persistente, o dispositivo não receberá mensagens enviadas enquanto o dispositivo não estiver conectado. Qual opção usar depende do aplicativo e se as mensagens que ocorrem enquanto um dispositivo não está conectado devem ser comunicadas. Para obter mais informações, consulte Usar sessões persistentes do MQTT (p. 88).

Qualidade de serviço

Quando o dispositivo publica e assina mensagens, a Qualidade de Serviço (QoS) preferida pode ser definida.AWS IoTsuporta QoS níveis 0 e 1 para operações de publicação e assinatura. Para obter mais informações sobre os níveis de QoS noAWS IoT, consulteOpções de qualidade de serviço (QoS) MQTT (p. 88).

OAWSO tempo de execução CRT para Python define essas constantes para os níveis de QoS que ele suporta:

Níveis de qualidade de serviço do Python

Nível de QoS	Valor simbólico Python usado pelo SDK	Descrição
QoS nível 0	`mqtt.QoS.AT_MOST_ONCE`	Apenas uma tentativa de enviar a mensagem será feita, seja ela recebida ou não. A mensagem pode não ser enviada, por exemplo, se o dispositivo não estiver conectado ou se houver um erro de rede.
QoS de nível 1	`mqtt.QoS.AT_LEAST_ONCE`	A mensagem é enviada repetidamente até um`PUBACK`confirmação é recebida.

No aplicativo de amostra, as solicitações de publicação e assinatura são feitas com um nível de QoS 1 (`mqtt.QoS.AT_LEAST_ONCE`).

- QoS na publicação

  Quando um dispositivo publica uma mensagem com QoS nível 1, ele envia a mensagem repetidamente até receber um`PUBACK`Resposta do agente de mensagens. Se o dispositivo não estiver conectado, a mensagem será enfileirada para ser enviada após a reconexão.
- QoS na assinatura

  Quando um dispositivo se inscreve em uma mensagem com QoS nível 1, o corretor de mensagens salva as mensagens nas quais o dispositivo está inscrito até que elas possam ser enviadas para o dispositivo. O corretor de mensagens reenvia as mensagens até receber um`PUBACK`resposta do dispositivo.

Publicar mensagens

Depois de estabelecer uma conexão comAWS IoT Core, os dispositivos podem publicar mensagens. O`pubsub.py`sample faz isso chamando o`publish`operação do`mqtt_connection`objeto.

```
mqtt_connection.publish(
    topic=args.topic,
    payload=message,
    qos=mqtt.QoS.AT_LEAST_ONCE
)
```

`topic`

O nome do tópico da mensagem que identifica a mensagem

No aplicativo de amostra, isso é transmitido pela linha de comando.

`payload`

A carga útil da mensagem formatada como uma string (por exemplo, um documento JSON)

No aplicativo de amostra, isso é transmitido pela linha de comando.

Um documento JSON é um formato de carga útil comum e um reconhecido por outroAWS IoTserviços; no entanto, o formato de dados da carga útil da mensagem pode ser qualquer coisa que os editores e assinantes concordem. Other (Outros)AWS IoTos serviços, no entanto, só reconhecem JSON e CBOR, em alguns casos, para a maioria das operações.

`qos`

O nível de QoS para esta mensagem

Assinatura de

Para receber mensagens doAWS IoTe outros serviços e dispositivos, os dispositivos assinam essas mensagens pelo nome do tópico. Os dispositivos podem se inscrever em mensagens individuais especificando umNome do tópico (p. 99)e para um grupo de mensagens especificando umFiltro de tópicos (p. 100), que pode incluir caracteres curinga. O`pubsub.py`sample usa o código mostrado aqui para assinar mensagens e registrar as funções de retorno de chamada para processar a mensagem depois que ela for recebida.

```
subscribe_future, packet_id = mqtt_connection.subscribe(
    topic=args.topic,
    qos=mqtt.QoS.AT_LEAST_ONCE,
    callback=on_message_received
)
subscribe_result = subscribe_future.result()
```

`topic`

O tópico que deseja assinar. Isso pode ser um nome de tópico ou um filtro de tópicos.

No aplicativo de amostra, isso é transmitido pela linha de comando.

`qos`

Se o corretor de mensagens deve armazenar essas mensagens enquanto o dispositivo estiver desconectado.

Um valor de`mqtt.QoS.AT_LEAST_ONCE`(QoS nível 1), requer que uma sessão persistente seja especificada (`clean_session=False`) quando a conexão é criada.

`callback`

A função a ser chamada para processar a mensagem inscrita.

O`mqtt_connection.subscribe`função retorna um futuro e um ID de pacote. Se a solicitação de assinatura tiver sido iniciada com êxito, o ID do pacote retornado será maior que 0. Para se certificar de que a assinatura foi recebida e registrada pelo corretor de mensagens, você deve aguardar o resultado da operação assíncrona retornar, conforme mostrado no exemplo de código.

A função de retorno de chamada

O retorno de chamada no`pubsub.py`sample processa as mensagens inscritas à medida que o dispositivo as recebe.

```
def on_message_received(topic, payload, **kwargs):
    print("Received message from topic '{}': {}".format(topic, payload))
    global received_count
    received_count += 1
    if received_count == args.count:
        received_all_event.set()
```

`topic`

O tópico da mensagem

Este é o nome de tópico específico da mensagem recebida, mesmo se você se inscreveu em um filtro de tópicos.

`payload`

A carga da mensagem

O formato para isso é específico do aplicativo.

`kwargs`

Possíveis argumentos adicionais conforme descrito em`mqtt.Connection.subscribe`.

No`pubsub.py`Exemplo,`on_message_received`exibe apenas o tópico e sua carga útil. Ele também conta as mensagens recebidas para encerrar o programa após o limite ser atingido.

Seu aplicativo avaliaria o tópico e a carga útil para determinar quais ações executar.

Desconexão e reconexão do dispositivo

O`pubsub.py`sample inclui funções de retorno de chamada que são chamadas quando o dispositivo é desconectado e quando a conexão é restabelecida. Quais ações seu dispositivo executa nesses eventos são específicas do aplicativo.

Quando um dispositivo se conecta pela primeira vez, ele deve se inscrever em tópicos para receber. Se a sessão de um dispositivo estiver presente quando ele se reconectar, suas assinaturas serão restauradas e todas as mensagens armazenadas dessas assinaturas serão enviadas para o dispositivo depois que ele se reconecta.

Se a sessão de um dispositivo não existir mais quando ele se reconecta, ele deverá ressubscrever para suas assinaturas. As sessões persistentes têm uma vida útil limitada e podem expirar quando o dispositivo é desconectado por muito tempo.

Connect o dispositivo e comunique comAWS IoT Core

Esta seção apresenta alguns exercícios para ajudá-lo a explorar diferentes aspectos da conexão do dispositivo aoAWS IoT Core. Para esses exercícios, você usará oCliente de teste MQTTnoAWS

IoTconsole para ver o que seu dispositivo publica e publicar mensagens no dispositivo. Esses exercícios usam o`pubsub.py`Exemplo doAWS IoTSDK de dispositivo v2 para Pythone aproveite sua experiência comConceitos básicos do AWS IoT Core (p. 18)tutoriais.

Nesta seção, você:

- Assine os filtros de tópicos curinga (p. 185)
- Assinaturas de filtro de tópicos de processo (p. 186)
- Publique mensagens do seu dispositivo (p. 188)

Para esses exercícios, você começará a partir do`pubsub.py`programa de amostra.

> Note
>
> Esses exercícios assumem que você completou oConceitos básicos do AWS IoT Core (p. 18)tutoriais e use a janela do terminal para seu dispositivo a partir desse tutorial.

Assine os filtros de tópicos curinga

Neste exercício, você modificará a linha de comando usada para chamar`pubsub.py`para se inscrever em um filtro de tópico curinga e processar as mensagens recebidas com base no tópico da mensagem.

Processo de exercício

Para este exercício, imagine que seu dispositivo contém um controle de temperatura e um controle de luz. Ele usa esses nomes de tópicos para identificar as mensagens sobre eles.

1. Antes de iniciar o exercício, tente executar este comando a partir doConceitos básicos do AWS IoT Core (p. 18)tutoriais em seu dispositivo para garantir que tudo esteja pronto para o exercício.

```
cd ~/aws-iot-device-sdk-python-v2/samples
python3 pubsub.py --topic topic_1 --ca_file ~/certs/Amazon-root-CA-1.pem --cert ~/
certs/device.pem.crt --key ~/certs/private.pem.key --endpoint your-iot-endpoint
```

   Você deve ver a mesma saída que você viu noTutorial de conceitos básicos (p. 61).

2. Para este exercício, altere esses parâmetros de linha de comando.

Ação	Parâmetro de linha de comando	Efeito
adicionar	--message ""	Configure`pubsub.py`para ouvir apenas
adicionar	--count 2	Termine o programa depois de receber duas mensagens
alteração	--topic `device/+/details`	Defina o filtro de tópicos para se inscrever

   Fazer essas alterações na linha de comando inicial resulta nesta linha de comando. Digite este comando na janela do terminal do seu dispositivo.

```
python3 pubsub.py --message "" --count 2 --topic device/+/details --ca_file ~/certs/
Amazon-root-CA-1.pem --cert ~/certs/device.pem.crt --key ~/certs/private.pem.key --
endpoint your-iot-endpoint
```

O programa deve exibir algo do tipo:

```
Connecting to a3qexamplesffp-ats.iot.us-west-2.amazonaws.com with client ID
 'test-24d7cdcc-cc01-458c-8488-2d05849691e1'...
Connected!
Subscribing to topic 'device/+/details'...
Subscribed with QoS.AT_LEAST_ONCE
Waiting for all messages to be received...
```

Se você vir algo assim em seu terminal, seu dispositivo estará pronto e ouvindo mensagens em que os nomes dos tópicos começam com`topic-1/`e termine com`/detail`. Então, vamos testar isso.

3. Aqui estão algumas mensagens que seu dispositivo pode receber.

Nome do tópico	Carga útil da mensagem
`device/temp/details`	`{ "desiredTemp": 20, "currentTemp": 15 }`
`device/light/details`	`{ "desiredLight": 100, "currentLight": 50 }`

4. usar o cliente de teste MQTT noAWS IoTconsole, envie as mensagens descritas na etapa anterior para o dispositivo.

   a. Abrir oCliente de teste MQTTnoAWS IoTconsole do .
   b. DentroAssinar um tópico, noCampo de tópico de assinatura, insira o filtro de tópicos:**`device/+/details`**e, depois, escolhaAssinar o tópico.
   c. NoAssinaturasColuna do cliente de teste MQTT, escolhadispositivo/+/detalhes.
   d. Para cada um dos tópicos na tabela anterior, faça o seguinte no cliente de teste MQTT:

      1. DentroPublicar, insira o valor daNome do tópicoColuna na tabela.
      2. No campo de carga útil da mensagem abaixo do nome do tópico, insira o valor doCarga útil da mensagemColuna na tabela.
      3. Assista à janela do terminal onde`pubsub.py`está em execução e, no cliente de teste MQTT, escolhaPublicar em um tópico.

      Você deve ver que a mensagem foi recebida por`pubsub.py`Na janela do terminal.

Resultado do exercício

Com isso,`pubsub.py`, inscreveu-se nas mensagens usando um filtro de tópico curinga, recebeu-as e as exibiu na janela do terminal. Observe como você se inscreveu em um único filtro de tópico e a função de retorno de chamada foi chamada para processar mensagens com dois tópicos distintos.

Assinaturas de filtro de tópicos de processo

Com base no exercício anterior, modifique o`pubsub.py`aplicativo de amostra para avaliar os tópicos da mensagem e processar as mensagens inscritas com base no tópico.

Processo de exercício

Para avaliar o tópico da mensagem

1. Copie `pubsub.py` para `pubsub2.py`.

2. Aberto`pubsub2.py`no seu editor de texto favorito ou IDE.
3. Dentro`pubsub2.py`, encontre `oon_message_received`função.
4. Dentro`on_message_received`, insira o código a seguir após a linha que começa com`print("Received message`e antes da linha que começa com`global received_count`.

```
    topic_parsed = False
    if "/" in topic:
        parsed_topic = topic.split("/")
        if len(parsed_topic) == 3:
            # this topic has the correct format
            if (parsed_topic[0] == 'device') and (parsed_topic[2] == 'details'):
                # this is a topic we care about, so check the 2nd element
                if (parsed_topic[1] == 'temp'):
                    print("Received temperature request: {}".format(payload))
                    topic_parsed = True
                if (parsed_topic[1] == 'light'):
                    print("Received light request: {}".format(payload))
                    topic_parsed = True
    if not topic_parsed:
        print("Unrecognized message topic.")
```

5. Salve suas alterações e execute o programa modificado usando esta linha de comando.

```
python3 pubsub2.py --message "" --count 2 --topic device/+/details --ca_file ~/certs/
Amazon-root-CA-1.pem --cert ~/certs/device.pem.crt --key ~/certs/private.pem.key --
endpoint your-iot-endpoint
```

6. NoAWS IoTconsole, abra oCliente de teste MQTT.
7. DentroAssinar um tópico, noCampo de tópico de assinatura, insira o filtro de tópicos:**`device/+/details`**e, depois, escolhaAssinar o tópico.
8. NoAssinaturasColuna do cliente de teste MQTT, escolhadispositivo/+/detalhes.
9. Para cada um dos tópicos desta tabela, faça o seguinte no cliente de teste MQTT:

Nome do tópico	Carga útil da mensagem
`device/temp/details`	`{ "desiredTemp": 20, "currentTemp": 15 }`
`device/light/details`	`{ "desiredLight": 100, "currentLight": 50 }`

   1. DentroPublicar, insira o valor daNome do tópicoColuna na tabela.
   2. No campo de carga útil da mensagem abaixo do nome do tópico, insira o valor doCarga útil da mensagemColuna na tabela.
   3. Assista à janela do terminal onde`pubsub.py`está em execução e, no cliente de teste MQTT, escolhaPublicar em um tópico.

   Você deve ver que a mensagem foi recebida por`pubsub.py`Na janela do terminal.

Será algo do tipo na janela do terminal.

```
Connecting to a3qexamplesffp-ats.iot.us-west-2.amazonaws.com with client ID 'test-
af794be0-7542-45a0-b0af-0b0ea7474517'...
Connected!
```

```
Subscribing to topic 'device/+/details'...
Subscribed with QoS.AT_LEAST_ONCE
Waiting for all messages to be received...
Received message from topic 'device/light/details': b'{ "desiredLight": 100,
 "currentLight": 50 }'
Received light request: b'{ "desiredLight": 100, "currentLight": 50 }'
Received message from topic 'device/temp/details': b'{ "desiredTemp": 20, "currentTemp":
 15 }'
Received temperature request: b'{ "desiredTemp": 20, "currentTemp": 15 }'
2 message(s) received.
Disconnecting...
Disconnected!
```

Resultado do exercício

Neste exercício, você adicionou código para que o aplicativo de amostra reconheça e processasse várias mensagens na função de retorno de chamada. Com isso, seu dispositivo pode receber mensagens e agir sobre elas.

Outra maneira de seu dispositivo receber e processar várias mensagens é assinar mensagens diferentes separadamente e atribuir cada assinatura à sua própria função de retorno de chamada.

Publique mensagens do seu dispositivo

Você pode usar o aplicativo de amostra pubsub.py para publicar mensagens do seu dispositivo. Embora ele publique mensagens como estão, as mensagens não podem ser lidas como documentos JSON. Este exercício modifica o aplicativo de amostra para poder publicar documentos JSON na carga útil da mensagem que pode ser lida porAWS IoT Core.

Processo de exercício

Neste exercício, a seguinte mensagem será enviada com `odevice/data`tópico.

```
{
    "timestamp": 1601048303,
    "sensorId": 28,
    "sensorData": [
        {
        "sensorName": "Wind speed",
        "sensorValue": 34.2211224
        }
    ]
}
```

Para preparar seu cliente de teste MQTT para monitorar as mensagens deste exercício

1. DentroAssinar um tópico, noCampo de tópico de assinatura, insira o filtro de tópicos:**`device/data`**e, depois, escolhaAssinar o tópico.
2. NoAssinaturasColuna do cliente de teste MQTT, escolhadispositivo/dados.
3. Mantenha a janela do cliente de teste MQTT aberta para aguardar mensagens do seu dispositivo.

Para enviar documentos JSON com o aplicativo de amostra pubsub.py

1. No dispositivo, copie`pubsub.py`para`pubsub3.py`.
2. Edite`pubsub3.py`para alterar a forma como ele formata as mensagens que publica.

    a. Em um editor de texto, abra `pubsub3.py`.

b. Localize essa linha de código:

```
message = "{} [{}]".format(args.message, publish_count)
```

c. Altere para:

```
message = "{}".format(args.message)
```

d. Salve as alterações.

3. No dispositivo, execute este comando para enviar a mensagem duas vezes.

```
python3 pubsub3.py  --ca_file ~/certs/Amazon-root-CA-1.pem  --cert ~/certs/
device.pem.crt  --key ~/certs/private.pem.key  --topic device/data  --count 2 --
message '{"timestamp":1601048303,"sensorId":28,"sensorData":[{"sensorName":"Wind
 speed","sensorValue":34.2211224}]}'  --endpoint your-iot-endpoint
```

4. No cliente de teste MQTT, verifique se ele interpretou e formatou o documento JSON na carga útil da mensagem, como esta:

Por padrão,`pubsub3.py`também se inscreve nas mensagens enviadas. Você deve ver que ele recebeu as mensagens na saída do aplicativo. A janela do terminal deve ser semelhante a este.

```
Connecting to a3qEXAMPLEsffp-ats.iot.us-west-2.amazonaws.com with client ID
 'test-5cff18ae-1e92-4c38-a9d4-7b9771afc52f'...
Connected!
Subscribing to topic 'device/data'...
Subscribed with QoS.AT_LEAST_ONCE
Sending 2 message(s)
Publishing message to topic 'device/data':
 {"timestamp":1601048303,"sensorId":28,"sensorData":[{"sensorName":"Wind
 speed","sensorValue":34.2211224}]}
Received message from topic 'device/data':
 b'{"timestamp":1601048303,"sensorId":28,"sensorData":[{"sensorName":"Wind
 speed","sensorValue":34.2211224}]}'
Publishing message to topic 'device/data':
 {"timestamp":1601048303,"sensorId":28,"sensorData":[{"sensorName":"Wind
 speed","sensorValue":34.2211224}]}
Received message from topic 'device/data':
 b'{"timestamp":1601048303,"sensorId":28,"sensorData":[{"sensorName":"Wind
 speed","sensorValue":34.2211224}]}'
2 message(s) received.
Disconnecting...
Disconnected!
```

Resultado do exercício

Com isso, seu dispositivo pode gerar mensagens para enviar paraAWS IoT Corepara testar a conectividade básica e fornecer mensagens de dispositivo paraAWS IoT CorePara processar. Por

exemplo, você pode usar este aplicativo para enviar dados de teste do seu dispositivo para testarAWS IoTAções de regra.

Analise os resultados

Os exemplos deste tutorial lhe deram experiência prática com os conceitos básicos de como os dispositivos podem se comunicar comAWS IoT Core—uma parte fundamental do seuAWS IoTsolução. Quando seus dispositivos conseguem se comunicar comAWS IoT Core, eles podem passar mensagens paraAWSserviços e outros dispositivos nos quais eles podem atuar. Da mesma forma,AWSserviços e outros dispositivos podem processar informações que resultam em mensagens enviadas de volta aos seus dispositivos.

Quando estiver pronto para explorarAWS IoT Corealém disso, experimente estes tutoriais:

- the section called “Enviando notificação do Amazon SNS” (p. 201)
- the section called “Armazenando dados do dispositivo em uma tabela do DynamoDB” (p. 209)
- the section called “Formatando uma notificação usando umAWS Lambdafunção” (p. 214)

Tutorial: Usar a AWS IoT Device SDK para C incorporado

Esta seção descreve como executar o AWS IoT Device SDK para C incorporado.

Procedimentos nesta seção

- Etapa1: Instalar a AWS IoT Device SDK para C incorporado (p. 190)
- Etapa 2: Configurar o aplicativo de amostra (p. 191)
- Etapa 3: Criar e executar a aplicação de exemplo (p. 192)

Etapa1: Instalar a AWS IoT Device SDK para C incorporado

Em geral, o AWS IoT Device SDK para C incorporado destina-se a dispositivos com restrição de recursos que exigem um tempo de execução de linguagem C otimizado. É possível usar o SDK em qualquer sistema operacional e hospedá-lo em qualquer tipo de processador (p. ex., MCUs e MPUs). Se você tiver mais memória e recursos de processamento disponíveis, recomendamos usar uma das ordens superioresAWS IoTSDKs de dispositivos e dispositivos móveis (por exemplo, C++, Java,JavaScripte Python).

Em geral, o AWS IoT Device SDK para C incorporado destina-se a sistemas que usam MCUs ou MPUs low-end que executam sistemas operacionais incorporados. Para o exemplo de programação nesta seção, assumimos que seu dispositivo usa o Linux.

Example

1. Fazer download doAWS IoT Device SDK para C incorporadopara o seu dispositivo a partir deGitHub.

   ```
   git clone https://github.com/aws/aws-iot-device-sdk-embedded-c.git --recurse-submodules
   ```

   Isso cria um diretório chamado `aws-iot-device-sdk-embedded-c` no diretório atual.
2. Navegue até esse diretório e faça o checkout da versão mais recente. Consultegithub.com/aws/aws-iot-device-SDK incorporadas-C/etiquetasPara obter a tag de versão mais recente.

   ```
   cd aws-iot-device-sdk-embedded-c
   git checkout latest-release-tag
   ```

3. Instalar o OpenSSL versão 1.1.0 ou posterior. As bibliotecas de desenvolvimento OpenSSL geralmente são chamadas de “libssl-dev” ou “openssl-devel” quando instaladas por meio de um gerenciador de pacotes.

```
sudo apt-get install libssl-dev
```

Etapa 2: Configurar o aplicativo de amostra

O AWS IoT Device SDK para C incorporado inclui aplicativos de exemplo para você experimentar. Para simplificar, este tutorial usa `amqtt_demo_mutual_auth`aplicativo, que ilustra como se conectar aoAWS IoT CoreAgente de mensagens e se inscreva e publique nos tópicos MQTT.

1. Copie o certificado e a chave privada que você criou noConceitos básicos do AWS IoT Core (p. 18)para o`build/bin/certificates`diretório.

   Note

   Os certificados CA raiz e de dispositivo estão sujeitos a expiração ou revogação. Se esses certificados expirarem ou forem revogados, será necessário copiar um novo certificado da CA ou uma chave privada e um certificado do dispositivo no seu dispositivo.

2. É necessário configurar o exemplo com sua chave privada, certificado, certificado CA raiz e endpoint pessoal do AWS IoT Core. Navegue até o diretório `aws-iot-device-sdk-embedded-c/demos/mqtt/mqtt_demo_mutual_auth`.

   Se você tiver aAWS CLIInstalado, você pode usar este comando para localizar a URL do endpoint da conta.

```
aws iot describe-endpoint --endpoint-type iot:Data-ATS
```

   Se você não tiver a AWS CLI instalada, abra o console do AWS IoT. No painel de navegação, escolha Manage (Gerenciar) e Things (Coisas). Escolha a coisa de IoT do seu dispositivo e escolhaInteragir. Seu endpoint é exibido na seção HTTPS da página de detalhes da coisa.

3. Abrir o`demo_config.h`Arquivo e atualize os valores do para o seguinte:

   AWS_IOT_ENDPOINT

   Seu endpoint pessoal.

   CLIENT_CERT_PATH

   O caminho do arquivo de certificado, por exemplo`certificates/device.pem.crt"`.

   CLIENT_PRIVATE_KEY_PATH

   O nome do arquivo de sua chave privada, por exemplo`certificates/private.pem.key`.

   Por exemplo:

```
// Get from demo_config.h
// ================================================
#define AWS_IOT_ENDPOINT                "my-endpoint-ats.iot.us-east-1.amazonaws.com"
#define AWS_MQTT_PORT                   8883
#define CLIENT_IDENTIFIER               "testclient"
#define ROOT_CA_CERT_PATH               "certificates/AmazonRootCA1.crt"
#define CLIENT_CERT_PATH                "certificates/my-device-cert.pem.crt"
#define CLIENT_PRIVATE_KEY_PATH         "certificates/my-device-private-key.pem.key"
// ================================================
```

4. Verifique se você tem o CMake instalado no seu dispositivo usando este comando.

```
cmake --version
```

Se vir as informações de versão para o compilador, você poderá continuar para a próxima seção.

Se receber um erro ou não vir nenhuma informação, você precisará instalar o pacote cmake usando este comando.

```
sudo apt-get install cmake
```

Execute acmake --versionComando novamente e confirme se o CMake foi instalado e que você está pronto para continuar.

5. Verifique se você tem as ferramentas de desenvolvimento instaladas em seu dispositivo usando este comando.

```
gcc --version
```

Se vir as informações de versão para o compilador, você poderá continuar para a próxima seção.

Se receber um erro ou não vir nenhuma informação do compilador, você deverá instalar o pacote `build-essential` usando este comando.

```
sudo apt-get install build-essential
```

Execute o comando gcc --version novamente e confirme se as ferramentas de compilação foram instaladas e que você está pronto para continuar.

Etapa 3: Criar e executar a aplicação de exemplo

Como executar os aplicativos de exemplo do AWS IoT Device SDK para C incorporado

1. Navegue até`aws-iot-device-sdk-embedded-c`e crie um diretório de compilação.

```
mkdir build && cd build
```

2. Digite o seguinte comando CMake para gerar os Makefiles necessários para construir.

```
cmake ..
```

3. Digite o comando a seguir para compilar o arquivo executável do aplicativo.

```
make
```

4. Execute o aplicativo `mqtt_demo_mutual_auth` com este comando.

```
cd bin
./mqtt_demo_mutual_auth
```

Você deve ver uma saída semelhante a:

```
[INFO] [DEMO] [mqtt_demo_mutual_auth.c:584] Establishing a TLS session to a2zk5tjv9x07ct-ats.iot.us-west-2.amazonaws.com:8883.
[INFO] [DEMO] [mqtt_demo_mutual_auth.c:1264] Creating an MQTT connection to a2zk5tjv9x07ct-ats.iot.us-west-2.amazonaws.com.
[INFO] [MQTT] [core_mqtt.c:855] Packet received. ReceivedBytes=2.
[INFO] [MQTT] [core_mqtt_serializer.c:970] CONNACK session present bit not set.
[INFO] [MQTT] [core_mqtt_serializer.c:912] Connection accepted.
[INFO] [MQTT] [core_mqtt.c:1526] Received MQTT CONNACK successfully from broker.
[INFO] [MQTT] [core_mqtt.c:1792] MQTT connection established with the broker.
[INFO] [DEMO] [mqtt_demo_mutual_auth.c:1033] MQTT connection successfully established with broker.


[INFO] [DEMO] [mqtt_demo_mutual_auth.c:1296] A clean MQTT connection is established. Cleaning up all the stored outgoing publishes.


[INFO] [DEMO] [mqtt_demo_mutual_auth.c:1314] Subscribing to the MQTT topic testclient/example/topic.
[INFO] [DEMO] [mqtt_demo_mutual_auth.c:1097] SUBSCRIBE sent for topic testclient/example/topic to broker.


[INFO] [MQTT] [core_mqtt.c:855] Packet received. ReceivedBytes=3.
[INFO] [DEMO] [mqtt_demo_mutual_auth.c:921] Subscribed to the topic testclient/example/topic. with maximum QoS 1.


[INFO] [DEMO] [mqtt_demo_mutual_auth.c:1358] Sending Publish to the MQTT topic testclient/example/topic.
[INFO] [DEMO] [mqtt_demo_mutual_auth.c:1195] PUBLISH sent for topic testclient/example/topic to broker with packet ID 2.


[INFO] [MQTT] [core_mqtt.c:855] Packet received. ReceivedBytes=2.
[INFO] [MQTT] [core_mqtt.c:1126] Ack packet deserialized with result: MQTTSuccess.
[INFO] [MQTT] [core_mqtt.c:1139] State record updated. New state=MQTTPublishDone.
[INFO] [DEMO] [mqtt_demo_mutual_auth.c:946] PUBACK received for packet id 2.


[INFO] [DEMO] [mqtt_demo_mutual_auth.c:672] Cleaned up outgoing publish packet with packet id 2.


[INFO] [MQTT] [core_mqtt.c:855] Packet received. ReceivedBytes=40.
[INFO] [MQTT] [core_mqtt.c:1015] De-serialized incoming PUBLISH packet: DeserializerResult=MQTTSuccess.
```

O dispositivo está conectado aoAWS IoTUsando aAWS IoT Device SDK para C incorporado.

Você também pode usar aAWS IoTPara visualizar as mensagens MQTT que o aplicativo de exemplo está publicando. Para obter informações sobre como usar o cliente MQTT no console do AWS IoT, consulte the section called “Visualizar mensagens MQTT com o cliente MQTT do AWS IoT” (p. 66).

CriarAWS IoTregras para rotear dados do dispositivo para outros serviços

Esses tutoriais mostram como criar e testarAWS IoTregras usando algumas das ações de regras mais comuns.

AWS IoTregras enviam dados de seus dispositivos para outrosAWSServiços da . Eles escutam mensagens MQTT específicas, formatam os dados nas cargas de mensagens e enviam o resultado para outrosAWSServiços da .

Recomendamos que você os experimente na ordem em que são mostrados aqui, mesmo que seu objetivo seja criar uma regra que use uma função do Lambda ou algo mais complexo. Os tutoriais são apresentados em ordem do básico ao complexo. Eles apresentam novos conceitos incrementalmente para ajudá-lo a aprender os conceitos que você pode usar para criar as ações de regras que não têm um tutorial específico.

> **Note**
>
> AWS IoTregras ajudam você a enviar os dados de seus dispositivos IoT para outrosAWSServiços da . Para fazer isso com sucesso, no entanto, você precisa de um conhecimento prático dos outros serviços para onde deseja enviar dados. Embora esses tutoriais forneçam as informações necessárias para concluir as tarefas, você pode achar útil saber mais sobre os serviços para os quais deseja enviar dados antes de usá-los em sua solução. Uma explicação detalhada do outroAWSos serviços estão fora do escopo desses tutoriais.

Visão geral do cenário do tutorial

O cenário desses tutoriais é o de um dispositivo sensor meteorológico que publica periodicamente seus dados. Existem muitos desses dispositivos de sensor neste sistema imaginário. Os tutoriais desta seção, no entanto, se concentram em um único dispositivo enquanto mostram como você pode acomodar vários sensores.

Os tutoriais nesta seção mostram como usarAWS IoTregras para fazer as seguintes tarefas com este sistema imaginário de dispositivos de sensores meteorológicos.

- Tutorial: Republicando uma mensagem MQTT (p. 195)

  Este tutorial mostra como republicar uma mensagem MQTT recebida dos sensores meteorológicos como uma mensagem que contém apenas o ID do sensor e o valor da temperatura. Ele usa apenasAWS IoT CoreServiços e demonstra uma consulta SQL simples e como usar o cliente MQTT do para testar a regra.
- Tutorial: Enviando notificação do Amazon SNS (p. 201)

  Este tutorial mostra como enviar uma mensagem SNS quando um valor de um dispositivo de sensor meteorológico excede um valor específico. Ele se baseia nos conceitos apresentados no tutorial anterior e adiciona como trabalhar com outroAWSserviço, oAmazon Simple Notification Service(Amazon SNS).

  Se você é novo com o Amazon SNS, consulte oConceitos básicosexercícios antes de iniciar este tutorial.
- Tutorial: Armazenando dados do dispositivo em uma tabela do DynamoDB (p. 209)

  Este tutorial mostra como armazenar os dados dos dispositivos do sensor meteorológico em uma tabela de banco de dados. Ele usa a instrução de consulta de regra e os modelos de substituição para formatar os dados da mensagem para o serviço de destino,Amazon DynamoDB.

  Se você é novo com o DynamoDB, revise oConceitos básicosexercícios antes de iniciar este tutorial.
- Tutorial: Formatando uma notificação usando umAWS Lambdafunção (p. 214)

  Este tutorial mostra como chamar uma função do Lambda para reformatar os dados do dispositivo e enviá-los como uma mensagem de texto. Ele adiciona um script do Python eAWSFunções do SDK em umAWS Lambdapara formatar com os dados de carga útil da mensagem dos dispositivos do sensor meteorológico e enviar uma mensagem de texto.

  Se você é novo com o Lambda, revise oConceitos básicosexercícios antes de iniciar este tutorial.

AWS IoTVisão geral da regra

Todos esses tutoriais criamAWS IoTregras.

Para umAWS IoTregra para enviar os dados de um dispositivo para outroAWSserviço, ele usa:

- Uma instrução de consulta de regra que consiste em:
  - Uma cláusula SQL SELECT que seleciona e formata os dados da carga útil da mensagem
  - Um filtro de tópico (o objeto FROM na instrução de consulta de regra) que identifica as mensagens a serem usadas
  - Uma instrução condicional opcional (uma cláusula SQL WHERE) que especifica condições específicas sobre as quais agir
- Pelo menos uma ação de regra

Os dispositivos publicam mensagens em tópicos MQTT. O filtro de tópico na instrução SQL SELECT identifica os tópicos do MQTT para aplicar a regra. Os campos especificados na instrução SQL SELECT formatam os dados da carga útil da mensagem MQTT recebida para uso pelas ações da regra. Para obter uma lista completa de ações de regra, consulte Ações de regra da AWS IoT (p. 485).

Tutoriais nesta seção

- Tutorial: Republicando uma mensagem MQTT (p. 195)
- Tutorial: Enviando notificação do Amazon SNS (p. 201)
- Tutorial: Armazenando dados do dispositivo em uma tabela do DynamoDB (p. 209)
- Tutorial: Formatando uma notificação usando umAWS Lambdafunção (p. 214)

Tutorial: Republicando uma mensagem MQTT

Este tutorial demonstra como criar umaAWS IoTregra que publica uma mensagem MQTT quando uma mensagem MQTT especificada é recebida. A carga útil da mensagem recebida pode ser modificada pela regra antes de ser publicada. Isso possibilita a criação de mensagens adaptadas a aplicativos específicos sem a necessidade de alterar o dispositivo ou o firmware. Você também pode usar o aspecto de filtragem de uma regra para publicar mensagens somente quando uma condição específica for atendida.

As mensagens republicadas por uma regra agem como mensagens enviadas por qualquer outraAWS IoTdispositivo ou cliente. Os dispositivos podem assinar as mensagens republicadas da mesma forma que podem se inscrever em qualquer outro tópico de mensagem MQTT.

O que você aprenderá neste tutorial:

- Como usar consultas e funções SQL simples em uma instrução de consulta de regra
- Como usar o cliente MQTT do para testar umAWS IoTregra

A conclusão deste tutorial requer cerca de 30 minutos.

Neste tutorial, você:

- Revise os tópicos do MQTT eAWS IoTregras (p. 195)
- Etapa 1: Criar umAWS IoTregra para republicar uma mensagem MQTT (p. 196)
- Etapa 2: Teste sua nova regra (p. 198)
- Etapa 3: Revise os resultados e as próximas etapas (p. 201)

Antes de começar este tutorial, verifique se você tem:

- Configurar seuConta da AWS (p. 19)

  Você precisará do seuConta da AWSeAWS IoTConsole para concluir este tutorial.
- AvaliadoVisualizar mensagens MQTT com o cliente MQTT do AWS IoT (p. 66)

  Certifique-se de que você pode usar o cliente MQTT do para se inscrever e publicar em um tópico. Você usará o cliente MQTT do para testar a nova regra neste procedimento.

Revise os tópicos do MQTT eAWS IoTregras

Antes de falar sobreAWS IoTregras, ajuda a entender o protocolo MQTT. Em soluções de IoT, o protocolo MQTT oferece algumas vantagens em relação a outros protocolos de comunicação de rede, como HTTP, o que o torna uma escolha popular para uso por dispositivos IoT. Esta seção analisa os principais aspectos do MQTT à medida que eles se aplicam a este tutorial. Para obter informações sobre como o MQTT se compara a HTTP, consulteEscolhendo um protocolo para a comunicação do seu dispositivo (p. 85).

Protocolo MQTT

O protocolo MQTT usa um modelo de comunicação de publicação/assinatura com o host. Para enviar dados, os dispositivos publicam mensagens identificadas por tópicos para oAWS IoTAgente de

mensagens. Para receber mensagens do corretor de mensagens, os dispositivos assinam os tópicos que receberão enviando filtros de tópicos em solicitações de assinatura para o corretor de mensagens. OAWS IoTO mecanismo de regras recebe mensagens MQTT do agente de mensagens.

Regras do AWS IoT

AWS IoTAs regras consistem em uma instrução de consulta de regra e uma ou mais ações de regra. Quando oAWS IoTo mecanismo de regras recebe uma mensagem MQTT, esses elementos atuam na mensagem da seguinte forma.

- Instrução de consulta de regras

  A instrução de consulta da regra descreve os tópicos MQTT a serem usados, interpreta os dados da carga útil da mensagem e formata os dados conforme descrito por uma instrução SQL semelhante às instruções usadas por bancos de dados SQL populares. O resultado da instrução de consulta são os dados enviados para as ações da regra.
- Ação da regra

  Cada ação de regra em uma regra atua nos dados resultantes da instrução de consulta da regra.AWS IoTaguentaMuitas ações de regra (p. 485). No entanto, neste tutorial, você se concentrará naRepublish (p. 546)ação de regra, que publica o resultado da instrução de consulta como uma mensagem MQTT com um tópico específico.

Etapa 1: Criar umAWS IoTregra para republicar uma mensagem MQTT

OAWS IoTregra que você criará neste tutorial se inscreve no`device/`*`device_id`*`/data`Tópicos MQTT em que*`device_id`*É o ID do dispositivo que enviou a mensagem. Esses tópicos são descritos por umFiltro de tópicos (p. 100)como`device/+/data`, em que o`+`É um caractere curinga que corresponde a qualquer string entre os dois caracteres de barra.

Quando a regra recebe uma mensagem de um tópico correspondente, ela republica `device_id`o`temperature`valores como uma nova mensagem MQTT com `device/data/temp`tópico.

Por exemplo, a carga útil de uma mensagem MQTT com `device/22/data`O tópico é semelhante a:

```
{
  "temperature": 28,
  "humidity": 80,
  "barometer": 1013,
  "wind": {
    "velocity": 22,
    "bearing": 255
  }
}
```

A regra toma o`temperature`valor da carga útil da mensagem e `device_id`do tópico, e os republica como uma mensagem MQTT com `device/data/temp`tópico e uma carga de mensagem semelhante a esta:

```
{
  "device_id": "22",
  "temperature": 28
}
```

Com essa regra, os dispositivos que precisam apenas do ID do dispositivo e os dados de temperatura assinam `device/data/temp`tópico para receber somente essas informações.

Para criar uma regra que republica uma mensagem MQTT

1. AbertoaRegrasHub doAWS IoTconsole.
2. DentroRegras, escolhaCriare comece a criar sua nova regra.
3. Na parte superior doCriar uma regra:
   a. DentroName (Nome), insira o nome da regra. Para este tutorial, nomeie-o**`republish_temp`**.

      Lembre-se de que um nome de regra deve ser exclusivo em sua Conta e Região e não pode ter espaços. Usamos um caractere sublinhado nesse nome para separar as duas palavras no nome da regra.
   b. DentroDescrição, descreva a regra.

      Uma descrição significativa ajuda você a lembrar o que essa regra faz e por que você a criou. A descrição pode ser o maior tempo necessário, portanto, seja o mais detalhado possível.
4. DentroInstrução de consulta de regrasdoCriar uma regra:
   a. DentroUso de versão SQL, selecione**`2016-03-23`**.
   b. NoInstrução de consulta de regrasCaixa de edição, insira a instrução:

      ```
      SELECT topic(2) as device_id, temperature FROM 'device/+/data'
      ```

      Esta afirmação:

      - Ouve mensagens MQTT com um tópico que corresponde ao`device/+/data`Filtro de tópicos.
      - Seleciona o segundo elemento da string de tópico e o atribui ao`device_id`campo.
      - Seleciona o valor`temperature`campo da carga útil da mensagem e o atribui à`temperature`campo.
5. DentroDefina uma ou mais ações:
   a. Para abrir a lista de ações de regras para essa regra, escolhaAção Adicionar ação.
   b. DentroSelecione uma ação, escolhaRepublique uma mensagem em umAWS IoTtópico.
   c. Na parte inferior da lista de ações, escolhaConfigure actionPara abrir a página de configuração da ação selecionada.
6. DentroConfigure action:
   a. DentroTópico, insira**`device/data/temp`**. Este é o tópico MQTT da mensagem que esta regra publicará.
   b. DentroQualidade de serviço, escolha0 - A mensagem é entregue zero ou mais vezes.
   c. DentroEscolha ou crie uma função para concederAWS IoTAcesso para executar essa ação:
      i. Selecione Create Role (Criar função). OCriar uma nova funçãoA caixa de diálogo é aberta.
      ii. Insira um nome que descreva a nova função. Neste tutorial, use**`republish_role`**.

         Quando você cria uma nova função, as políticas corretas para executar a ação da regra são criadas e anexadas à nova função. Se você alterar o tópico dessa ação de regra ou usar essa função em outra ação de regra, deverá atualizar a política dessa função para autorizar o novo tópico ou ação. Para atualizar uma função existente, escolhaAtualizar funçãoNesta seção.
      iii. SelecioneCriar funçãoPara criar a função e fechar a caixa de diálogo.
   d. SelecioneAção Adicionar açãopara adicionar a ação à regra e retornar aoCriar uma regra.
7. ORepublique uma mensagem em umAWS IoTtópicoação agora está listada emDefina uma ou mais ações.

No bloco da nova ação, abaixoRepublique uma mensagem em umAWS IoTtópico, você pode ver o tópico no qual sua ação de republicação será publicada.

Esta é a única ação de regra que você adicionará a essa regra.

8. DentroCriar uma regra, role para baixo até a parte inferior e escolhaCriar regrapara criar a regra e concluir esta etapa.

Etapa 2: Teste sua nova regra

Para testar sua nova regra, você usará o cliente MQTT para publicar e assinar as mensagens MQTT usadas por esta regra.

Abrir oCliente MQTT noAWS IoTconsoleEm uma nova janela. Isso permitirá que você edite a regra sem perder a configuração do seu cliente MQTT. O cliente MQTT não retém assinaturas ou registros de mensagens se você deixar ele ir para outra página no console.

Para usar o cliente MQTT do para testar a regra

1. NoCliente MQTT noAWS IoTconsole, assine os tópicos de entrada, neste caso,`device/+/data`.
   a. No Cliente MQTT, emAssinaturas, escolhaAssinar um tópico.
   b. DentroTópico de assinatura, insira o tópico do filtro de tópico de entrada,**`device/+/data`**.
   c. Mantenha o restante dos campos nas configurações padrão.
   d. Escolha Subscribe to topic (Inscrever-se no tópico).

      NoAssinaturascoluna, sobPublicar em um tópico,**`device/+/data`**é exibida.
2. Assine o tópico que sua regra publicará:`device/data/temp`.
   a. UNDERAssinaturas, escolhaAssinar um tópiconovamente, e emTópico de assinatura, insira o tópico da mensagem republicada,**`device/data/temp`**.
   b. Mantenha o restante dos campos nas configurações padrão.
   c. Escolha Subscribe to topic (Inscrever-se no tópico).

      NoAssinaturascoluna, sobdispositivo/+/dados,**`device/data/temp`**é exibida.
3. Publique uma mensagem no tópico de entrada com um ID de dispositivo específico,**`device/22/data`**. Você não pode publicar em tópicos do MQTT que contenham caracteres curinga.
   a. No Cliente MQTT, emAssinaturas, escolhaPublicar em um tópico.
   b. NoPublicar, insira o nome do tópico de entrada,**`device/22/data`**.
   c. Copie os dados de amostra mostrados aqui e, na caixa de edição abaixo do nome do tópico, cole os dados de amostra.

```
{
  "temperature": 28,
  "humidity": 80,
  "barometer": 1013,
  "wind": {
    "velocity": 22,
    "bearing": 255
  }
}
```

   d. Para enviar sua mensagem MQTT, escolhaPublicar em um tópico.
4. Revise as mensagens enviadas.

a. No Cliente MQTT, emAssinaturas, há um ponto verde ao lado dos dois tópicos nos quais você se inscreveu anteriormente.

   Os pontos verdes indicam que uma ou mais novas mensagens foram recebidas desde a última vez que você as examinou.

b. UNDERAssinaturas, escolhadispositivo/+/dadospara verificar se a carga útil da mensagem corresponde ao que você acabou de publicar e se parece com isso:

```
{
  "temperature": 28,
  "humidity": 80,
  "barometer": 1013,
  "wind": {
    "velocity": 22,
    "bearing": 255
  }
}
```

c. UNDERAssinaturas, escolhadispositivo/dados/temporáriopara verificar se a carga de mensagem republicada se parece com isso:

```
{
  "device_id": "22",
  "temperature": 28
}
```

   Observe que o`device_id`value é uma string entre aspas e o`temperature`valor é numérico. Isso ocorre porque o`topic()`a função extraiu a string do nome do tópico da mensagem de entrada enquanto o`temperature`value usa o valor numérico da carga útil da mensagem de entrada.

   Se você quiser fazer`device_id`valor um valor numérico, substitua`topic(2)`Na instrução de consulta de regra com:

```
cast(topic(2) AS DECIMAL)
```

   Note que lançando o`topic(2)`valor para um valor numérico só funcionará se essa parte do tópico contiver apenas caracteres numéricos.

5. Se você perceber que a mensagem correta foi publicada nodispositivo/dados/temporáriotópico, então sua regra funcionou. Veja o que mais você pode aprender sobre a ação Republicar regra na próxima seção.

   Se você não vir que a mensagem correta foi publicada para qualquer um dosdispositivo/+/dadosoudispositivo/dados/temporáriotópicos, verifique as dicas de solução de problemas.

Solucionando problemas de sua regra de mensagem de republicação

Aqui estão algumas coisas para verificar caso você não esteja vendo os resultados esperados.

- Você recebeu um banner de erro

  Se um erro aparecer quando você publicou a mensagem de entrada, corrija esse erro primeiro. As etapas a seguir podem ajudá-lo a corrigir esse erro.
- A mensagem de entrada não será exibida no MQTT cliente do

Toda vez que você publica sua mensagem de entrada no`device/22/data`tópico, essa mensagem deve aparecer no cliente MQTT se você se inscreveu no`device/+/data`Filtro de tópicos como descrito no procedimento.

Coisas para verificar

- Verifique o filtro de tópicos no qual você se inscreveu

  Se você se inscreveu no tópico da mensagem de entrada conforme descrito no procedimento, verá uma cópia da mensagem de entrada toda vez que a publicar.

  Se você não vir a mensagem, verifique o nome do tópico que você se inscreveu e compare-o com o tópico no qual você publicou. Os nomes de tópicos diferenciam maiúsculas de minúsculas e o tópico no qual você se inscreveu deve ser idêntico ao tópico ao qual você publicou a carga útil da mensagem.
- Verifique a função de publicação de mensagens

  No Cliente MQTT, emAssinaturas, escolhadispositivo/+/dados, verifique o tópico da mensagem de publicação e, em seguida, escolhaPublicar em um tópico. Você deve ver a carga útil da mensagem na caixa de edição abaixo do tópico aparecer na lista de mensagens.

- Você não vê sua mensagem republicada no MQTT cliente do

  Para que sua regra funcione, ela deve ter a política correta que a autorize a receber e republicar uma mensagem e deve receber a mensagem.

  ### Coisas para verificar

  - Verifique oRegião da AWSdo seu cliente MQTT e da regra que você criou

    O console no qual você está executando o cliente MQTT deve estar no mesmoAWSRegião como a regra que você criou.
  - Verifique o tópico da mensagem de entrada na instrução de consulta de regra

    Para que a regra funcione, ela deve receber uma mensagem com o nome do tópico que corresponde ao filtro de tópico na cláusula FROM da instrução de consulta de regra.

    Verifique a ortografia do filtro de tópico na instrução de consulta de regra com a do tópico no cliente MQTT. Os nomes de tópicos diferenciam maiúsculas e minúsculas e o tópico da mensagem deve corresponder ao filtro de tópico na instrução de consulta de regra
  - Verifique o conteúdo da carga da mensagem de entrada

    Para que a regra funcione, ela deve encontrar o campo de dados na carga da mensagem declarada na instrução SELECT.

    Verifique a ortografia do`temperature`campo na instrução de consulta de regra com a da carga útil da mensagem no cliente MQTT. Os nomes dos campos diferenciam letras maiúsculas de`temperature`No comando de consulta de regra deve ser idêntico ao`temperature`campo na carga útil da mensagem.

    Certifique-se de que o documento JSON na carga da mensagem esteja formatado corretamente. Se o JSON tiver algum erro, como uma vírgula ausente, a regra não poderá lê-lo.
  - Verifique o tópico da mensagem republicada na ação da regra

    O tópico no qual a ação Republicar regra publica a nova mensagem deve corresponder ao tópico no qual você se inscreveu no cliente MQTT.

    Abra a regra que você criou no console e verifique o tópico para o qual a ação da regra republicará a mensagem.

- Verifique a função que está sendo usada pela regra

  A ação da regra deve ter permissão para receber o tópico original e publicar o novo tópico.

  As políticas que autorizam a regra a receber dados de mensagens e republicá-los são específicas dos tópicos usados. Se você alterar o tópico usado para republicar os dados da mensagem, deverá atualizar a função da ação da regra para atualizar sua política para corresponder ao tópico atual.

  Se você suspeitar que esse é o problema, edite a ação Republicar regra e crie uma nova função. Novas funções criadas pela ação de regra recebem as autorizações necessárias para executar essas ações.

Etapa 3: Revise os resultados e as próximas etapas

Neste tutorial

- Você usou uma consulta SQL simples e algumas funções em uma instrução de consulta de regra para produzir uma nova mensagem MQTT.
- Você criou uma regra que publicou novamente essa nova mensagem.
- Você usou o cliente MQTT do para testar oAWS IoTregra.

Next steps (Próximas etapas)

Depois de republicar algumas mensagens com essa regra, tente experimentar com ela para ver como a alteração de alguns aspectos do tutorial afeta a mensagem republicada. Veja aqui algumas ideias para começar a usar.

- Altere o`device_id`no tópico da mensagem de entrada e observe o efeito na carga de mensagem republicada.
- Altere os campos selecionados na instrução de consulta de regra e observe o efeito na carga de mensagem republicada.
- Experimente o próximo tutorial desta série e aprenda comoTutorial: Enviando notificação do Amazon SNS (p. 201).

A ação Republicar regra usada neste tutorial também pode ajudá-lo a depurar instruções de consulta de regra. Por exemplo, você pode adicionar essa ação a uma regra para ver como sua instrução de consulta de regra está formatando os dados usados por suas ações de regra.

Tutorial: Enviando notificação do Amazon SNS

Este tutorial demonstra como criar umaAWS IoTregra que envia dados de mensagem MQTT para um tópico do Amazon SNS para que ele possa ser enviado como uma mensagem de texto SMS.

Neste tutorial, você cria uma regra que envia dados de mensagem de um sensor meteorológico para todos os assinantes de um tópico do Amazon SNS, sempre que a temperatura excede o valor definido na regra. A regra detecta quando a temperatura relatada excede o valor definido pela regra e, em seguida, cria uma nova carga útil de mensagem que inclui apenas o ID do dispositivo, a temperatura relatada e o limite de temperatura que foi excedido. A regra envia a nova carga de mensagem como um documento JSON para um tópico do SNS, que notifica todos os assinantes sobre o tópico do SNS.

O que você aprenderá neste tutorial:

- Como criar e testar uma notificação do Amazon SNS
- Como chamar uma notificação do Amazon SNS de umAWS IoTregra

- Como usar consultas e funções SQL simples em uma instrução de consulta de regra
- Como usar o cliente MQTT do para testar umAWS IoTregra

A conclusão deste tutorial requer cerca de 30 minutos.

Neste tutorial, você:

- Etapa 1: Criar um tópico do Amazon SNS que envia uma mensagem de texto SMS (p. 202)
- Etapa 2: Criar umAWS IoTRegra para enviar a mensagem de texto (p. 203)
- Etapa 3: Teste oAWS IoTNotificação por regra e Amazon SNS (p. 205)
- Etapa 4: Revise os resultados e as próximas etapas (p. 208)

Antes de começar este tutorial, verifique se você tem:

- Configurar seuConta da AWS (p. 19)

  Você precisará do seuConta da AWSeAWS IoTConsole para concluir este tutorial.
- AvaliadoVisualizar mensagens MQTT com o cliente MQTT do AWS IoT (p. 66)

  Certifique-se de que você pode usar o cliente MQTT do para se inscrever e publicar em um tópico. Você usará o cliente MQTT do para testar a nova regra neste procedimento.
- Analise oAmazon Simple Notification Service

  Se você ainda não usou o Amazon SNS, consulteConfigurar o acesso para o Amazon SNS. Se você já concluiu outroAWS IoTtutoriais, seuConta da AWSjá deve estar configurado corretamente.

Etapa 1: Criar um tópico do Amazon SNS que envia uma mensagem de texto SMS

Para criar um tópico do Amazon SNS que envia uma mensagem de texto SMS

1. Crie um tópico do Amazon SNS.

   a. Faça login no console do Amazon SNS.
   b. No painel de navegação à esquerda, selecione Topics (Tópicos).
   c. Na página Topics (Tópicos), escolha Create topic (Criar tópico).
   d. DentroDetalhes, escolha oStandard (Padrão)type. Por padrão, o console cria um tópico FIFO.
   e. DentroName (Nome), insira o nome do tópico do SNS. Para este tutorial, insira **`high_temp_notice`**.
   f. Role até o final da página e escolhaCriar tópico.

      O console abrirá a página Details (Detalhes) do tópico.
2. Crie uma assinatura do Amazon SNS.

   > **Note**
   >
   > O número de telefone que você usa nesta assinatura pode incorrer em cobranças de mensagens de texto das mensagens que você enviará neste tutorial.

   a. Nohigh_temp_noticePágina de detalhes do tópico, escolhaCriar assinatura.
   b. DentroCriar assinatura, noDetalhesseção, na seçãoProtocoloLista, escolhaSMS.
   c. DentroPonto final, insira o número de um telefone que pode receber mensagens de texto. Certifique-se de inseri-lo de tal forma que comece com um+, inclui o país e o código de área e não inclui nenhum outro caractere de pontuação.

   d. Selecione Create subscription.
3. Teste a notificação do Amazon SNS.
   a. NoConsole do Amazon SNSNo painel de navegação à esquerda, escolhaTópicos.
   b. Para abrir a página de detalhes do tópico, emTópicosNa lista de tópicos, escolhahigh_temp_notice.
   c. Para abrir oPublicar mensagem no tópicoPágina, nohigh_temp_noticePágina de detalhes, escolhaPublicar mensagem.
   d. DentroPublicar mensagem no tópico, noCorpo da mensagemseção, noCorpo da mensagem a ser enviada para o endpoint, insira uma mensagem curta.
   e. Role para baixo até o final da página e selecionePublicar mensagem.
   f. No telefone com o número usado anteriormente ao criar a assinatura, confirme se a mensagem foi recebida.

   Se você não recebeu a mensagem de teste, verifique novamente o número de telefone e as configurações do telefone.

   Certifique-se de que você pode publicar mensagens de teste a partir doConsole do Amazon SNSAntes de continuar o tutorial.

Etapa 2: Criar umAWS IoTRegra para enviar a mensagem de texto

OAWS IoTregra que você criará neste tutorial se inscreve no`device/`*`device_id`*`/data`Tópicos MQTT em que*`device_id`*É o ID do dispositivo que enviou a mensagem. Esses tópicos são descritos em um filtro de tópicos como`device/+/data`, em que o`+`É um caractere curinga que corresponde a qualquer string entre os dois caracteres de barra. Esta regra também testa o valor do`temperature`campo na carga útil da mensagem.

Quando a regra recebe uma mensagem de um tópico correspondente, ela leva a*`device_id`*do nome do tópico, o`temperature`valor da carga útil da mensagem e adiciona um valor constante para o limite que está testando e envia esses valores como um documento JSON para um tópico de notificação do Amazon SNS.

Por exemplo, uma mensagem MQTT do dispositivo sensor meteorológico número 32 usa o`device/32/data`tópico e tem uma carga de mensagem semelhante a esta:

```
{
  "temperature": 38,
  "humidity": 80,
  "barometer": 1013,
  "wind": {
    "velocity": 22,
    "bearing": 255
  }
}
```

A instrução de consulta de regra da regra usa o`temperature`valor da carga útil da mensagem, o*`device_id`*do nome do tópico e adiciona a constante`max_temperature`valor para enviar uma carga de mensagem que se parece com esta para o tópico do Amazon SNS:

```
{
  "device_id": "32",
  "reported_temperature": 38,
  "max_temperature": 30
}
```

Para criar umAWS IoTregra para detectar um valor de temperatura de excesso de limite e criar os dados a serem enviados para o tópico do Amazon SNS

1. AbertoaRegrasHub doAWS IoTconsole.
2. Se esta for a sua primeira regra, escolhaCriar, ouCriar uma regra.
3. DentroCriar uma regra:
   a. Em Name (Nome), insira **`temp_limit_notify`**.

      Lembre-se de que o nome da regra deve ser exclusivo dentro daConta da AWSe Região, e não pode ter espaços. Usamos um caractere sublinhado nesse nome para separar as palavras no nome da regra.
   b. DentroDescrição, descreva a regra.

      Uma descrição significativa facilita lembrar o que essa regra faz e por que você a criou. A descrição pode ser o maior tempo necessário, portanto, seja o mais detalhado possível.
4. DentroInstrução de consulta de regrasdoCriar uma regra:
   a. DentroUso de versão SQL, selecione23/03/2016.
   b. NoInstrução de consulta de regrasCaixa de edição, insira a instrução:

```
SELECT topic(2) as device_id,
    temperature as reported_temperature,
    30 as max_temperature
  FROM 'device/+/data'
  WHERE temperature > 30
```

      Esta afirmação:

      - Ouve mensagens MQTT com um tópico que corresponde ao`device/+/data`filtro de tópicos e que têm um`temperature`valor maior que 30.
      - Seleciona o segundo elemento da string de tópico e o atribui ao`device_id`campo.
      - Seleciona o valor`temperature`campo da carga útil da mensagem e o atribui à`reported_temperature`campo.
      - Cria um valor constante`30`para representar o valor-limite e atribuí-lo ao`max_temperature`campo.
5. Para abrir a lista de ações de regras para essa regra, emConfigure uma ou mais ações, escolhaAção Adicionar ação.
6. DentroSelecione uma ação, escolhaEnvie uma mensagem como uma notificação por push do SNS.
7. Para abrir a página de configuração da ação selecionada, na parte inferior da lista de ações, escolhaConfigure action.
8. DentroConfigure action:
   a. DentroDestino do SNS, escolhaSelect, encontre seu tópico do SNShigh_temp_notice, e escolhaSelect.
   b. DentroFormato da mensagem, escolhaCRU.
   c. DentroEscolha ou crie uma função para concederAWS IoTAcesso para executar essa ação, escolhaCriar função.
   d. DentroCriar uma nova função, emName (Nome), insira um nome exclusivo para a nova função. Para este tutorial, use **`sns_rule_role`**.
   e. Selecione Create role (Criar função).

Se você estiver repetindo este tutorial ou reutilizando uma função existente, escolhaAtualizar funçãoantes de continuar. Isso atualiza o documento de política da função para trabalhar com o destino do SNS.

9. SelecioneAção Adicionar açãoe volte para oCriar uma regra.

   No bloco da nova ação, abaixoEnvie uma mensagem como uma notificação por push do SNS, você pode ver o tópico do SNS que sua regra chamará.

   Esta é a única ação de regra que você adicionará a essa regra.

10. Para criar a regra e concluir esta etapa, emCriar uma regra, role para baixo até a parte inferior e escolhaCriar regra.

Etapa 3: Teste oAWS IoTNotificação por regra e Amazon SNS

Para testar sua nova regra, você usará o cliente MQTT para publicar e assinar as mensagens MQTT usadas por esta regra.

Abrir oCliente MQTT noAWS IoTconsoleem uma nova janela. Isso permitirá que você edite a regra sem perder a configuração do seu cliente MQTT. Se você deixar o cliente MQTT para ir para outra página no console, ele não manterá nenhuma assinatura ou registro de mensagens.

Para usar o cliente MQTT do para testar a regra

1. NoCliente MQTT noAWS IoTconsole, assine os tópicos de entrada, neste caso,`device/+/data`.

   a. No Cliente MQTT, emAssinaturas, escolhaAssinar um tópico.

   b. DentroTópico de assinatura, insira o tópico do filtro de tópico de entrada,**`device/+/data`**.

   c. Mantenha o restante dos campos nas configurações padrão.

   d. Escolha Subscribe to topic (Inscrever-se no tópico).

      NoAssinaturascoluna, sobPublicar em um tópico,**`device/+/data`**é exibida.

2. Publique uma mensagem no tópico de entrada com um ID de dispositivo específico,**`device/32/data`**. Você não pode publicar em tópicos do MQTT que contenham caracteres curinga.

   a. No Cliente MQTT, emAssinaturas, escolhaPublicar em um tópico.

   b. NoPublicar, insira o nome do tópico de entrada,**`device/32/data`**.

   c. Copie os dados de amostra mostrados aqui e, na caixa de edição abaixo do nome do tópico, cole os dados de amostra.

```
{
  "temperature": 38,
  "humidity": 80,
  "barometer": 1013,
  "wind": {
    "velocity": 22,
    "bearing": 255
  }
}
```

   d. SelecionePublicar em um tópicopara publicar sua mensagem MQTT.

3. Confirme se a mensagem de texto foi enviada.

   a. No Cliente MQTT, emAssinaturas, há um ponto verde ao lado do tópico no qual você se inscreveu anteriormente.

O ponto verde indica que uma ou mais novas mensagens foram recebidas desde a última vez que você as examinou.

b. UNDERAssinaturas, escolhadispositivo/+/dadospara verificar se a carga útil da mensagem corresponde ao que você acabou de publicar e se parece com isso:

```
{
  "temperature": 38,
  "humidity": 80,
  "barometer": 1013,
  "wind": {
    "velocity": 22,
    "bearing": 255
  }
}
```

c. Verifique o telefone que você usou para assinar o tópico do SNS e confirme se o conteúdo da carga da mensagem é assim:

```
{"device_id":"32","reported_temperature":38,"max_temperature":30}
```

Observe que o`device_id`value é uma string entre aspas e o`temperature`valor é numérico. Isso ocorre porque o`topic()`a função extraiu a string do nome do tópico da mensagem de entrada enquanto o`temperature`value usa o valor numérico da carga útil da mensagem de entrada.

Se você quiser fazer o`device_id`valor um valor numérico, substitua`topic(2)`Na instrução de consulta de regra com:

```
cast(topic(2) AS DECIMAL)
```

Note que lançando o`topic(2)`valor para um numérico,`DECIMAL`value só funcionará se essa parte do tópico contiver apenas caracteres numéricos.

4. Tente enviar uma mensagem MQTT na qual a temperatura não exceda o limite.

   a. No Cliente MQTT, emAssinaturas, escolhaPublicar em um tópico.

   b. NoPublicar, insira o nome do tópico de entrada,**`device/33/data`**.

   c. Copie os dados de amostra mostrados aqui e, na caixa de edição abaixo do nome do tópico, cole os dados de amostra.

```
{
  "temperature": 28,
  "humidity": 80,
  "barometer": 1013,
  "wind": {
    "velocity": 22,
    "bearing": 255
  }
}
```

   d. Para enviar sua mensagem MQTT, escolhaPublicar em um tópico.

A mensagem que enviou na**`device/+/data`**Assinatura. No entanto, como o valor da temperatura está abaixo da temperatura máxima na instrução de consulta de regra, você não deve receber uma mensagem de texto.

Se não vir o comportamento correto, verifique as dicas de solução de problemas.

Solução de problemas da regra de mensagem do SNS

Aqui estão algumas coisas para verificar, caso você não esteja vendo os resultados esperados.

- Você recebeu um banner de erro

  Se um erro aparecer quando você publicou a mensagem de entrada, corrija esse erro primeiro. As etapas a seguir podem ajudá-lo a corrigir esse erro.

- A mensagem de entrada não será exibida no MQTT cliente do

  Toda vez que você publica sua mensagem de entrada no`device/22/data`tópico, essa mensagem deve aparecer no cliente MQTT, se você se inscreveu no`device/+/data`Filtro de tópicos como descrito no procedimento.

  ### Coisas para verificar

  - Verifique o filtro de tópicos no qual você se inscreveu

    Se você se inscreveu no tópico da mensagem de entrada conforme descrito no procedimento, verá uma cópia da mensagem de entrada toda vez que a publicar.

    Se você não vir a mensagem, verifique o nome do tópico que você se inscreveu e compare-o com o tópico no qual você publicou. Os nomes de tópicos diferenciam maiúsculas de minúsculas e o tópico no qual você se inscreveu deve ser idêntico ao tópico ao qual você publicou a carga útil da mensagem.

  - Verifique a função de publicação de mensagens

    No Cliente MQTT, emAssinaturas, escolhadispositivo/+/dados, verifique o tópico da mensagem de publicação e, em seguida, escolhaPublicar em um tópico. Você deve ver a carga útil da mensagem na caixa de edição abaixo do tópico aparecer na lista de mensagens.

- Você não recebe uma mensagem SMS

  Para que sua regra funcione, ela deve ter a política correta que a autoriza a receber uma mensagem e enviar uma notificação do SNS, e ela deve receber a mensagem.

  ### Coisas para verificar

  - Verifique oRegião da AWSdo seu cliente MQTT e da regra que você criou

    O console no qual você está executando o cliente MQTT deve estar no mesmoAWSRegião como a regra que você criou.

  - Verifique se o valor da temperatura na carga útil da mensagem excede o limite de teste

    Se o valor da temperatura for menor ou igual a 30, conforme definido na instrução de consulta de regra, a regra não executará nenhuma de suas ações.

  - Verifique o tópico da mensagem de entrada na instrução de consulta de regra

    Para que a regra funcione, ela deve receber uma mensagem com o nome do tópico que corresponde ao filtro de tópico na cláusula FROM da instrução de consulta de regra.

    Verifique a ortografia do filtro de tópico na instrução de consulta de regra com a do tópico no cliente MQTT. Os nomes de tópicos diferenciam maiúsculas e minúsculas e o tópico da mensagem deve corresponder ao filtro de tópico na instrução de consulta de regra

  - Verifique o conteúdo da carga da mensagem de entrada

    Para que a regra funcione, ela deve encontrar o campo de dados na carga da mensagem declarada na instrução SELECT.

Verifique a ortografia do`temperature`campo na instrução de consulta de regra com a da carga útil da mensagem no cliente MQTT. Os nomes dos campos diferenciam letras maiúsculas de`temperature`No comando de consulta de regra deve ser idêntico ao`temperature`campo na carga útil da mensagem.

Certifique-se de que o documento JSON na carga da mensagem esteja formatado corretamente. Se o JSON tiver algum erro, como uma vírgula ausente, a regra não poderá lê-lo.

- Verifique o tópico da mensagem republicada na ação da regra

  O tópico no qual a ação Republicar regra publica a nova mensagem deve corresponder ao tópico no qual você se inscreveu no cliente MQTT.

  Abra a regra que você criou no console e verifique o tópico para o qual a ação da regra republicará a mensagem.

- Verifique a função que está sendo usada pela regra

  A ação da regra deve ter permissão para receber o tópico original e publicar o novo tópico.

  As políticas que autorizam a regra a receber dados de mensagens e republicá-los são específicas dos tópicos usados. Se você alterar o tópico usado para republicar os dados da mensagem, deverá atualizar a função da ação da regra para atualizar sua política para corresponder ao tópico atual.

  Se você suspeitar que esse é o problema, edite a ação Republicar regra e crie uma nova função. Novas funções criadas pela ação de regra recebem as autorizações necessárias para executar essas ações.

Etapa 4: Revise os resultados e as próximas etapas

Neste tutorial:

- Você criou e testou um tópico de notificação do Amazon SNS e uma assinatura.
- Você usou uma consulta e funções SQL simples em uma instrução de consulta de regra para criar uma nova mensagem para sua notificação.
- Você criou umAWS IoTregra para enviar uma notificação do Amazon SNS que usou sua carga de mensagem personalizada.
- Você usou o cliente MQTT do para testar oAWS IoTregra.

Next steps (Próximas etapas)

Depois de enviar algumas mensagens de texto com essa regra, tente experimentar com ela para ver como a mudança de alguns aspectos do tutorial afeta a mensagem e quando ela é enviada. Veja aqui algumas ideias para começar a usar.

- Altere o`device_id`no tópico da mensagem de entrada e observe o efeito no conteúdo da mensagem de texto.
- Altere os campos selecionados na instrução de consulta de regra e observe o efeito no conteúdo da mensagem de texto.
- Altere o teste na instrução de consulta de regra para testar uma temperatura mínima em vez de uma temperatura máxima. Lembre-se de mudar o nome do`max_temperature`!
- Adicione uma ação de regra de republicação para enviar uma mensagem MQTT quando uma notificação do SNS for enviada.
- Experimente o próximo tutorial desta série e aprenda comoTutorial: Armazenando dados do dispositivo em uma tabela do DynamoDB (p. 209).

Tutorial: Armazenando dados do dispositivo em uma tabela do DynamoDB

Este tutorial demonstra como criar umaAWS IoTregra que envia dados da mensagem para uma tabela do DynamoDB.

Neste tutorial, você cria uma regra que envia dados de mensagem de um dispositivo de sensor meteorológico imaginário para uma tabela do DynamoDB. A regra formata os dados de muitos sensores meteorológicos, de modo que eles possam ser adicionados a uma única tabela de banco de dados.

O que você aprenderá neste tutorial

- Como criar uma tabela do DynamoDB
- Como enviar dados de mensagem para uma tabela do DynamoDB a partir de umaAWS IoTregra
- Como usar modelos de substituição em umAWS IoTregra
- Como usar consultas e funções SQL simples em uma instrução de consulta de regra
- Como usar o cliente MQTT do para testar umAWS IoTregra

A conclusão deste tutorial requer cerca de 30 minutos.

Neste tutorial, você:

- Etapa 1: Criar uma tabela do DynamoDB para este tutorial (p. 209)
- Etapa 2: Criar umAWS IoTregra para enviar dados para a tabela do DynamoDB (p. 210)
- Etapa 3: Teste oAWS IoTTabela do DynamoDB (p. 211)
- Etapa 4: Revise os resultados e as próximas etapas (p. 214)

Antes de começar este tutorial, verifique se você tem:

- Configurar seuConta da AWS (p. 19)

  Você precisará do seuConta da AWSeAWS IoTConsole para concluir este tutorial.
- AvaliadoVisualizar mensagens MQTT com o cliente MQTT do AWS IoT (p. 66)

  Certifique-se de que você pode usar o cliente MQTT do para se inscrever e publicar em um tópico. Você usará o cliente MQTT do para testar a nova regra neste procedimento.
- Analise oAmazon DynamoDBvisão geral

  Se você não usou o DynamoDB antes, reviseConceitos básicos do DynamoDBpara se familiarizar com os conceitos e operações básicos do DynamoDB.

Etapa 1: Criar uma tabela do DynamoDB para este tutorial

Neste tutorial, você criará uma tabela do DynamoDB com esses atributos para registrar os dados dos dispositivos de sensores meteorológicos imaginários:

- `sample_time`é uma chave primária e descreve a hora em que a amostra foi gravada.
- `device_id`é uma chave de classificação e descreve o dispositivo que forneceu a amostra
- `device_data`são os dados recebidos do dispositivo e formatados pela instrução de consulta de regra

Para criar uma tabela do DynamoDB para este tutorial

1. Abrir oConsole do DynamoDBe, depois, escolhaCriar tabela.

2. DentroCriar tabela:

   a. DentroNome da tabela, insira o nome da tabela:**`wx_data`**.
   b. DentroChave de partição, insira**`sample_time`**e na lista de opções ao lado do campo, escolha**`Number`**.
   c. DentroChave de classificação, insira**`device_id`**e na lista de opções ao lado do campo, escolha**`Number`**.
   d. Na parte inferior da página, escolhaCriar.

Você definirá`device_data`mais tarde, quando você configura a ação de regra do DynamoDB.

Etapa 2: Criar umAWS IoTregra para enviar dados para a tabela do DynamoDB

Nesta etapa, você usará a instrução de consulta de regra para formatar os dados dos dispositivos de sensor meteorológico imaginários para gravar na tabela do banco de dados.

Uma carga útil de mensagem de amostra recebida de um dispositivo de sensor meteorológico é assim:

```
{
  "temperature": 28,
  "humidity": 80,
  "barometer": 1013,
  "wind": {
    "velocity": 22,
    "bearing": 255
  }
}
```

Para a entrada do banco de dados, você usará a instrução de consulta de regra para nivelar a estrutura da carga útil da mensagem para ficar assim:

```
{
  "temperature": 28,
  "humidity": 80,
  "barometer": 1013,
  "wind_velocity": 22,
  "wind_bearing": 255
}
```

Nesta regra, você também usará algunsModelos de substituição (p. 627). Modelos de substituição são expressões que permitem inserir valores dinâmicos de funções e dados de mensagens.

Para criar oAWS IoTregra para enviar dados para a tabela do DynamoDB

1. Abertoo hub de regras doAWS IoTconsole.
2. Para começar a criar sua nova regra noRegras, escolhaCriar regra.
3. DentroPropriedades das regras:

   a. DentroRule name (Nome da regra), insira**`wx_data_ddb`**.

      Lembre-se de que o nome da regra deve ser exclusivo dentro daConta da AWSe Região, e não pode ter espaços. Usamos um caractere sublinhado nesse nome para separar as duas palavras no nome da regra.
   b. DentroDescrição da regra, descreva a regra.

      Uma descrição significativa facilita lembrar o que essa regra faz e por que você a criou. A descrição pode ser o maior tempo necessário, portanto, seja o mais detalhado possível.

4. Escolha Próximo para continuar.
5. DentroDeclaração do SQL:
   a. DentroVersão do SQL, selecione**`2016-03-23`**.
   b. NoDeclaração do SQLCaixa de edição, insira a instrução:

```
SELECT temperature, humidity, barometer,
  wind.velocity as wind_velocity,
  wind.bearing as wind_bearing,
FROM 'device/+/data'
```

   Esta afirmação:

   - Ouve mensagens MQTT com um tópico que corresponde ao`device/+/data`Filtro de tópicos.
   - Formata os elementos do`wind`atributo como atributos individuais.
   - Passagens no`temperature`,`humidity`, e`barometer`atributos inalterados.
6. Escolha Próximo para continuar.
7. DentroAções da regra:
   a. Para abrir a lista de ações de regras para essa regra, emAção 1, escolha**`DynamoDB`**.
   b. DentroNome da tabela, escolha o nome da tabela do DynamoDB criada em uma etapa anterior:**`wx_data`**.

      OTipo de chave de partiçãoeTipo de chave de classificaçãoos campos são preenchidos com os valores da tabela do DynamoDB.
   c. Em Partition key (Chave de partição), insira **`sample_time`**.
   d. Em Partition key value (Valor da chave de partição), insira **`${timestamp()}`**.

      Este é o primeiro doModelos de substituição (p. 627)você usará nesta regra. Em vez de usar um valor da carga útil da mensagem, ele usará o valor retornado da função de carimbo de data/hora. Para saber mais, consultetimestamp (p. 619)noAWS IoT CoreGuia do desenvolvedor.
   e. DentroChave de classificação, insira**`device_id`**.
   f. Em Sort key value (Valor da chave de classificação), insira **`${cast(topic(2) AS DECIMAL)}`**.

      Este é o segundo dosModelos de substituição (p. 627)você usará nesta regra. Ele insere o valor do segundo elemento no nome do tópico, que é o ID do dispositivo, depois de lançá-lo para um valor DECIMAL para corresponder ao formato numérico da chave. Para saber mais sobre tópicos, consultetópico (p. 619)noAWS IoT CoreGuia do desenvolvedor. Ou para saber mais sobre elenco, consultelançar (p. 587)noAWS IoT CoreGuia do desenvolvedor.
   g. Na coluna Write message data to this column (Gravar os dados da mensagem nesta coluna), insira **`device_data`**.

      Isso criará o`device_data`Na tabela do DynamoDB.
   h. Deixe Operation (Operação) em branco.
   i. DentroIAM role (Função do IAM), escolha**`wx_ddb_role`**.
   j. Na parte inferior da página, selecione Próximo.
8. Na parte inferior doRevisar e criarPágina, escolhaCriarPara criar a regra.

Etapa 3: Teste oAWS IoTTabela do DynamoDB

Para testar a nova regra, você usará o cliente MQTT para publicar e assinar as mensagens MQTT usadas neste teste.

Abrir oCliente MQTT noAWS IoTconsoleem uma nova janela. Isso permitirá que você edite a regra sem perder a configuração do seu cliente MQTT. O cliente MQTT não retém assinaturas ou registros de mensagens se você deixar ele ir para outra página no console. Você também vai querer que uma janela de console separada seja aberta para oHub de tabelas do DynamoDB noAWS IoTconsolepara exibir as novas entradas que sua regra envia.

Para usar o cliente MQTT do para testar a regra

1. NoCliente MQTT noAWS IoTconsole, assine o tópico de entrada,`device/+/data`.
   a. No Cliente MQTT, escolhaAssinar um tópico.
   b. para oFiltro de tópicos, insira o tópico do filtro de tópico de entrada,**`device/+/data`**.
   c. Escolha Subscribe.
2. Agora, publique uma mensagem no tópico de entrada com um ID de dispositivo específico,**`device/22/data`**. Você não pode publicar em tópicos do MQTT que contenham caracteres curinga.
   a. No Cliente MQTT, escolhaPublicar em um tópico.
   b. para oNome do tópico, insira o nome do tópico de entrada,**`device/22/data`**.
   c. para oCarga útil da mensagem, insira os dados de amostra a seguir.

   ```
   {
     "temperature": 28,
     "humidity": 80,
     "barometer": 1013,
     "wind": {
       "velocity": 22,
       "bearing": 255
     }
   }
   ```

   d. Para publicar a mensagem MQTT, escolhaPublicar.
   e. Agora, no Cliente MQTT, escolhaAssinar um tópico. NoSubscribecoluna, escolha a**`device/+/data`**Assinatura. Confirme se os dados de amostra da etapa anterior aparecem lá.
3. Verifique para ver a linha na tabela do DynamoDB que sua regra criou.
   a. NoHub de tabelas do DynamoDB noAWS IoTconsole, escolhawx_datae, depois, escolha oItensGuia.

      Se você já estiver noItens, Talvez seja necessário atualizar a exibição escolhendo o ícone de atualização no canto superior direito do cabeçalho da tabela.
   b. Observe que osample_timevalores na tabela são links e abrem um. Se você acabou de enviar sua primeira mensagem, ela será a única na lista.

      Esse link exibe todos os dados nessa linha da tabela.
   c. Expandir odevice_datapara ver os dados resultantes da instrução de consulta de regra.
   d. Explore as diferentes representações dos dados disponíveis nesta tela. Você também pode editar os dados nesta tela.
   e. Depois de terminar de revisar essa linha de dados, para salvar as alterações feitas, escolhaSalvar, ou para sair sem salvar nenhuma alteração, escolhaCancelar.

Se não vir o comportamento correto, verifique as dicas de solução de problemas.

Solucionar problemas da regra do DynamoDB

Aqui estão algumas coisas para verificar caso você não esteja vendo os resultados esperados.

- Você recebeu um banner de erro

  Se um erro aparecer quando você publicou a mensagem de entrada, corrija esse erro primeiro. As etapas a seguir podem ajudá-lo a corrigir esse erro.

- A mensagem de entrada não será exibida no MQTT cliente do

  Toda vez que você publica sua mensagem de entrada no`device/22/data`tópico, essa mensagem deve aparecer no cliente MQTT se você se inscreveu no`device/+/data`Filtro de tópicos como descrito no procedimento.

  Coisas para verificar

  - Verifique o filtro de tópicos no qual você se inscreveu

    Se você se inscreveu no tópico da mensagem de entrada conforme descrito no procedimento, verá uma cópia da mensagem de entrada toda vez que a publicar.

    Se você não vir a mensagem, verifique o nome do tópico que você se inscreveu e compare-o com o tópico no qual você publicou. Os nomes de tópicos diferenciam maiúsculas de minúsculas e o tópico no qual você se inscreveu deve ser idêntico ao tópico ao qual você publicou a carga útil da mensagem.

  - Verifique a função de publicação de mensagens

    No Cliente MQTT, emAssinaturas, escolhadispositivo/+/dados, verifique o tópico da mensagem de publicação e, em seguida, escolhaPublicar em um tópico. Você deve ver a carga útil da mensagem na caixa de edição abaixo do tópico aparecer na lista de mensagens.

- Você não vê seus dados na tabela do DynamoDB

  A primeira coisa a fazer é atualizar a exibição escolhendo o ícone de atualização no canto superior direito do cabeçalho da tabela. Se isso não exibir os dados que você está procurando, verifique o seguinte.

  Coisas para verificar

  - Verifique oRegião da AWSdo seu cliente MQTT e da regra que você criou

    O console no qual você está executando o cliente MQTT deve estar no mesmoAWSRegião como a regra que você criou.

  - Verifique o tópico da mensagem de entrada na instrução de consulta de regra

    Para que a regra funcione, ela deve receber uma mensagem com o nome do tópico que corresponde ao filtro de tópico na cláusula FROM da instrução de consulta de regra.

    Verifique a ortografia do filtro de tópico na instrução de consulta de regra com a do tópico no cliente MQTT. Os nomes de tópicos diferenciam maiúsculas e minúsculas e o tópico da mensagem deve corresponder ao filtro de tópico na instrução de consulta de regra

  - Verifique o conteúdo da carga da mensagem de entrada

    Para que a regra funcione, ela deve encontrar o campo de dados na carga da mensagem declarada na instrução SELECT.

    Verifique a ortografia do`temperature`campo na instrução de consulta de regra com a da carga útil da mensagem no cliente MQTT. Os nomes dos campos diferenciam letras maiúsculas de`temperature`No comando de consulta de regra deve ser idêntico ao`temperature`campo na carga útil da mensagem.

    Certifique-se de que o documento JSON na carga da mensagem esteja formatado corretamente. Se o JSON tiver algum erro, como uma vírgula ausente, a regra não poderá lê-lo.

- Verifique os nomes de chave e campo usados na ação da regra

  Os nomes de campo usados na regra de tópico devem corresponder aos encontrados na carga útil da mensagem JSON da mensagem publicada.

  Abra a regra que você criou no console e verifique os nomes dos campos na configuração da ação da regra com os usados no cliente MQTT.
- Verifique a função que está sendo usada pela regra

  A ação da regra deve ter permissão para receber o tópico original e publicar o novo tópico.

  As políticas que autorizam a regra a receber dados de mensagens e atualizar a tabela do DynamoDB são específicas dos tópicos usados. Se você alterar o tópico ou o nome da tabela do DynamoDB usado pela regra, deverá atualizar a função da ação da regra para atualizar sua política para corresponder.

  Se você suspeitar que esse é o problema, edite a ação da regra e crie uma nova função. Novas funções criadas pela ação de regra recebem as autorizações necessárias para executar essas ações.

Etapa 4: Revise os resultados e as próximas etapas

Depois de enviar algumas mensagens para a tabela do DynamoDB com essa regra, tente experimentar com ela para ver como a alteração de alguns aspectos do tutorial afeta os dados gravados na tabela. Veja aqui algumas ideias para começar a usar.

- Altere o`device_id`no tópico da mensagem de entrada e observe o efeito sobre os dados. Você pode usar isso para simular dados de recebimento de vários sensores meteorológicos.
- Altere os campos selecionados na instrução de consulta de regra e observe o efeito sobre os dados. É possível usar isso para filtrar os dados armazenados na tabela.
- Adicione uma ação de regra de republicação para enviar uma mensagem MQTT para cada linha adicionada à tabela. Você poderia usar isso para depuração.

Depois de concluir este tutorial, confirathe section called “Formatando uma notificação usando umAWS Lambdafunção” (p. 214).

Tutorial: Formatando uma notificação usando umAWS Lambdafunção

Este tutorial demonstra como enviar dados de mensagem MQTT do para umAWS Lambdaação para formatação e envio para outroAWSserviço. Neste tutorial, oAWS LambdaA ação usa a açãoAWSSDK para enviar a mensagem formatada para o tópico do Amazon SNS que você criou no tutorial sobre comothe section called “Enviando notificação do Amazon SNS” (p. 201).

No tutorial sobre comothe section called “Enviando notificação do Amazon SNS” (p. 201), o documento JSON que resultou da instrução de consulta da regra foi enviado como o corpo da mensagem de texto. O resultado foi uma mensagem de texto que se parecia com este exemplo:

```
{"device_id":"32","reported_temperature":38,"max_temperature":30}
```

Neste tutorial, você usará umAWS Lambdaação de regra para chamar umAWS Lambdafunção que formata os dados da instrução de consulta de regra em um formato mais amigável, como este exemplo:

```
Device 32 reports a temperature of 38, which exceeds the limit of 30.
```

OAWS Lambdafunção que você criará neste tutorial formata a string de mensagem usando os dados da instrução de consulta de regra e chama oPublicação do SNSFunção doAWSSDK para criar a notificação.

O que você aprenderá neste tutorial

- Como criar e testar umAWS Lambdafunção
- Como usar oAWSSDK em umAWS Lambdafunção para publicar uma notificação do Amazon SNS
- Como usar consultas e funções SQL simples em uma instrução de consulta de regra
- Como usar o cliente MQTT do para testar umAWS IoTregra

A conclusão deste tutorial requer cerca de 45 minutos.

Neste tutorial, você:

- Etapa 1: Criar umAWS LambdaFunção que envia mensagens de texto (p. 215)
- Etapa 2: Criar umAWS IoTregra com umAWS LambdaAção da regra do (p. 217)
- Etapa 3: Teste oAWS IoTregra eAWS LambdaAção da regra do (p. 219)
- Etapa 4: Revise os resultados e as próximas etapas (p. 222)

Antes de começar este tutorial, verifique se você tem:

- Configurar seuConta da AWS (p. 19)

  Você precisará do seuConta da AWSeAWS IoTConsole para concluir este tutorial.
- AvaliadoVisualizar mensagens MQTT com o cliente MQTT do AWS IoT (p. 66)

  Certifique-se de que você pode usar o cliente MQTT do para se inscrever e publicar em um tópico. Você usará o cliente MQTT do para testar a nova regra neste procedimento.
- Concluiu os outros tutoriais de regras nesta seção

  Este tutorial requer o tópico de notificação do SNS que você criou no tutorial sobre comothe section called “Enviando notificação do Amazon SNS” (p. 201). Também pressupõe que você tenha concluído os outros tutoriais relacionados a regras nesta seção.
- Analise oAWS Lambdavisão geral

  Se você ainda não usouAWS Lambdaantes, revejaAWS LambdaeConceitos básicos do LambdaPara aprender seus termos e conceitos.

Etapa 1: Criar umAWS LambdaFunção que envia mensagens de texto

OAWS Lambdaneste tutorial recebe o resultado da instrução de consulta de regra, insere os elementos em uma string de texto e envia a string resultante para o Amazon SNS como a mensagem em uma notificação.

Ao contrário do tutorial sobre comothe section called “Enviando notificação do Amazon SNS” (p. 201), que usou umAWS IoTação de regra para enviar a notificação, este tutorial envia a notificação da função do Lambda usando uma função doAWSSDK. O tópico de notificação real do Amazon SNS usado neste tutorial, no entanto, é o mesmo que você usou no tutorial sobre comothe section called “Enviando notificação do Amazon SNS” (p. 201).

Para criar umAWS LambdaFunção que envia mensagens de texto

1. Criar um novoAWS Lambdafunção.

   a. No console do AWS Lambda, escolha Create function (Criar função).
   b. DentroCriar função, selecioneUse um blueprint.

Pesquise e selecione o**`hello-world-python`**blueprint e, em seguida, escolhaConfigure.

c. DentroInformações básicas:

   i. DentroNome da função, insira o nome dessa função,**`format-high-temp-notification`**.

   ii. DentroFunção de execução, escolhaCriar uma nova função a partir deAWSModelos de política.

   iii. Em Role name (Nome da função), insira o nome da nova função,**`format-high-temp-notification-role`**.

   iv. DentroModelos de política -opcionais, procure e selecionePolítica de publicação do Amazon SNS.

   v. Escolha Create function (Criar função).

2. Modifique o código do blueprint para formatar e enviar uma notificação do Amazon SNS.

   a. Depois de criar sua função, você deve ver anotificação em formato de alta temperaturaPágina de detalhes. Se você não fizer isso, abra-o a partir doLambdaFunções.

   b. Nonotificação em formato de alta temperaturaPágina de detalhes, escolhaConfiguraçãoguia e role até a guiaCódigo da funçãoPainel.

   c. NoCódigo da funçãojanela, noAmbiente, escolha o arquivo Python,`lambda_function.py`.

   d. NoCódigo da função, exclua todo o código do programa original do blueprint e substitua-o por este código.

```
import boto3
#
#   expects event parameter to contain:
#   {
#       "device_id": "32",
#       "reported_temperature": 38,
#       "max_temperature": 30,
#       "notify_topic_arn": "arn:aws:sns:us-east-1:57EXAMPLE833:high_temp_notice"
#   }
#
#   sends a plain text string to be used in a text message
#
#      "Device {0} reports a temperature of {1}, which exceeds the limit of {2}."
#
#   where:
#       {0} is the device_id value
#       {1} is the reported_temperature value
#       {2} is the max_temperature value
#
def lambda_handler(event, context):

    # Create an SNS client to send notification
    sns = boto3.client('sns')

    # Format text message from data
    message_text = "Device {0} reports a temperature of {1}, which exceeds the
 limit of {2}.".format(
            str(event['device_id']),
            str(event['reported_temperature']),
            str(event['max_temperature'])
        )

    # Publish the formatted message
    response = sns.publish(
            TopicArn = event['notify_topic_arn'],
            Message = message_text
        )
```

```
    return response
```

e. Escolha Implantar.

3. Em uma nova janela, procure o nome do recurso da Amazon (ARN) do tópico do Amazon SNS do tutorial sobre comothe section called “Enviando notificação do Amazon SNS” (p. 201).

   a. Em uma nova janela, abra aPágina de tópicos do console do Amazon SNS.
   b. NoTópicosPágina, localize ohigh_temp_noticetópico de notificação na lista de tópicos do Amazon SNS.
   c. Encontre oARNdohigh_temp_noticetópico de notificação a ser usado na próxima etapa.

4. Crie um caso de teste para sua função Lambda.

   a. NoLambdaFunçõesPágina do console, nanotificação em formato de alta temperaturaPágina de detalhes, escolhaSelecione um evento de testeNo canto superior direito da página (mesmo que pareça desativado) e, em seguida, escolhaConfigure test events.
   b. DentroConfigure test event, escolhaCriar novo evento de teste.
   c. DentroEvent name (Nome do evento), insira**`SampleRuleOutput`**.
   d. No editor JSON abaixoEvent name (Nome do evento), cole este exemplo de documento JSON. Este é um exemplo do queAWS IoTregra será enviada para a função do Lambda.

```
{
  "device_id": "32",
  "reported_temperature": 38,
  "max_temperature": 30,
  "notify_topic_arn": "arn:aws:sns:us-east-1:57EXAMPLE833:high_temp_notice"
}
```

   e. Consulte a janela que tem oARNdohigh_temp_noticetópico de notificação e copie o valor de ARN.
   f. Substitua `onotify_topic_arn`valor no editor JSON com o ARN do tópico de notificação.

      Mantenha esta janela aberta para que você possa usar esse valor de ARN novamente ao criar oAWS IoTregra.
   g. Escolha Create (Criar).

5. Teste a função com dados de amostra.

   a. Nonotificação em formato de alta temperaturaNa página de detalhes, no canto superior direito da página, confirme queSampleRuleOutputaparece ao lado doTesteBotão. Caso contrário, escolhe-o na lista de eventos de teste disponíveis.
   b. Para enviar a mensagem de saída da regra de amostra para sua função, escolhaTeste.

Se a função e a notificação funcionarem, você receberá uma mensagem de texto no telefone que se inscreveu na notificação.

Se você não recebeu uma mensagem de texto no telefone, verifique o resultado da operação. NoCódigo da funçãopainel, noResultado da execução, revise a resposta para encontrar quaisquer erros ocorridos. Não prossiga para a próxima etapa até que a função possa enviar a notificação ao telefone.

Etapa 2: Criar umAWS IoTregra com umAWS LambdaAção da regra do

Nesta etapa, você usará a instrução de consulta de regra para formatar os dados do dispositivo de sensor meteorológico imaginário para enviar para uma função do Lambda, que formatará e enviará uma mensagem de texto.

Uma carga útil de mensagem de amostra recebida dos dispositivos meteorológicos é assim:

```
{
```

```
  "temperature": 28,
  "humidity": 80,
  "barometer": 1013,
  "wind": {
    "velocity": 22,
    "bearing": 255
  }
}
```

Nesta regra, você usará a instrução de consulta de regra para criar uma carga de mensagem para a função do Lambda que se parece com isso:

```
{
  "device_id": "32",
  "reported_temperature": 38,
  "max_temperature": 30,
  "notify_topic_arn": "arn:aws:sns:us-east-1:57EXAMPLE833:high_temp_notice"
}
```

Isso contém todas as informações que a função Lambda precisa para formatar e enviar a mensagem de texto correta.

Para criar oAWS IoTregra para chamar uma função do Lambda

1. Abrir oRegrasHub doAWS IoTconsole.
2. Para começar a criar sua nova regra noRegras, escolhaCriar.
3. Na parte superior doCriar uma regra:
   a. DentroName (Nome), insira o nome da regra,**`wx_friendly_text`**.

      Lembre-se de que o nome da regra deve ser exclusivo dentro daConta da AWSe Região, e não pode ter espaços. Usamos um caractere sublinhado nesse nome para separar as duas palavras no nome da regra.
   b. DentroDescrição, descreva a regra.

      Uma descrição significativa facilita lembrar o que essa regra faz e por que você a criou. A descrição pode ser o maior tempo necessário, portanto, seja o mais detalhado possível.
4. DentroInstrução de consulta de regrasdoCriar uma regra:
   a. DentroUso de versão SQL, selecione**`2016-03-23`**.
   b. NoInstrução de consulta de regrasCaixa de edição, insira a instrução:

      ```
      SELECT
        cast(topic(2) AS DECIMAL) as device_id,
        temperature as reported_temperature,
        30 as max_temperature,
        'arn:aws:sns:us-east-1:57EXAMPLE833:high_temp_notice' as notify_topic_arn
      FROM 'device/+/data' WHERE temperature > 30
      ```

      Esta afirmação:

      - Ouve mensagens MQTT com um tópico que corresponde ao`device/+/data`filtro de tópicos e que têm um`temperature`valor maior que 30.
      - Seleciona o segundo elemento da string de tópico, converte-o em um número decimal e, em seguida, o atribui ao`device_id`campo.
      - Seleciona o valor do`temperature`campo da carga útil da mensagem e o atribui à`reported_temperature`campo.

- Cria um valor constante,`30`, para representar o valor-limite e atribuí-lo a`max_temperature`campo.
- Cria um valor constante para `notify_topic_arn`campo.

c. Consulte a janela que tem oARNdohigh_temp_noticetópico de notificação e copie o valor de ARN.

d. Substitua o valor ARN (*`arn:aws:sns:us-east- 1:57 exemplo833:HIGH_TEMP_NOTICE`*) no editor de instruções de consulta de regras com o ARN do tópico de notificação.

5. DentroConfigure uma ou mais ações:

   a. Para abrir a lista de ações de regras para essa regra, escolhaAção Adicionar ação.

   b. DentroSelecione uma ação, escolhaEnvie uma mensagem para uma função do Lambda.

   c. Para abrir a página de configuração da ação selecionada, na parte inferior da lista de ações, escolhaConfigure action.

6. DentroConfigure action:

   a. DentroNome da função, escolhaSelect.

   b. Selecionenotificação em formato de alta temperatura.

   c. Na parte inferior doConfigure action, escolhaAção Adicionar ação.

   d. Para criar a regra, na parte inferior doCriar uma regra, escolhaCriar regra.

Etapa 3: Teste oAWS IoTregra eAWS LambdaAção da regra do

Para testar sua nova regra, você usará o cliente MQTT para publicar e assinar as mensagens MQTT usadas por esta regra.

Abrir oCliente MQTT noAWS IoTconsoleem uma nova janela. Agora você pode editar a regra sem perder a configuração do seu cliente MQTT. Se você deixar o cliente MQTT para ir para outra página no console, perderá suas assinaturas ou registros de mensagens.

Para usar o cliente MQTT do para testar a regra

1. NoCliente MQTT noAWS IoTconsole, assine os tópicos de entrada, neste caso,`device/+/data`.

   a. No Cliente MQTT, emAssinaturas, escolhaAssinar um tópico.

   b. DentroTópico de assinatura, insira o tópico do filtro de tópico de entrada,**`device/+/data`**.

   c. Mantenha o restante dos campos nas configurações padrão.

   d. Escolha Subscribe to topic (Inscrever-se no tópico).

   NoAssinaturascoluna, sobPublicar em um tópico,**`device/+/data`**é exibida.

2. Publique uma mensagem no tópico de entrada com um ID de dispositivo específico,**`device/32/data`**. Você não pode publicar em tópicos do MQTT que contenham caracteres curinga.

   a. No Cliente MQTT, emAssinaturas, escolhaPublicar em um tópico.

   b. NoPublicar, insira o nome do tópico de entrada,**`device/32/data`**.

   c. Copie os dados de amostra mostrados aqui e, na caixa de edição abaixo do nome do tópico, cole os dados de amostra.

```
{
  "temperature": 38,
  "humidity": 80,
  "barometer": 1013,
  "wind": {
    "velocity": 22,
    "bearing": 255
  }
```

```
}
```

d. Para publicar sua mensagem MQTT, escolhaPublicar em um tópico.

3. Confirme se a mensagem de texto foi enviada.

   a. No Cliente MQTT, emAssinaturas, há um ponto verde ao lado do tópico no qual você se inscreveu anteriormente.

      O ponto verde indica que uma ou mais novas mensagens foram recebidas desde a última vez que você as examinou.

   b. UNDERAssinaturas, escolhadispositivo/+/dadospara verificar se a carga útil da mensagem corresponde ao que você acabou de publicar e se parece com isso:

```
{
  "temperature": 38,
  "humidity": 80,
  "barometer": 1013,
  "wind": {
    "velocity": 22,
    "bearing": 255
  }
}
```

   c. Verifique o telefone que você usou para assinar o tópico do SNS e confirme se o conteúdo da carga da mensagem é assim:

```
Device 32 reports a temperature of 38, which exceeds the limit of 30.
```

      Se você alterar o elemento ID do tópico no tópico da mensagem, lembre-se de lançar `otopic(2)`valor para um valor numérico só funcionará se esse elemento no tópico da mensagem contiver apenas caracteres numéricos.

4. Tente enviar uma mensagem MQTT na qual a temperatura não exceda o limite.

   a. No Cliente MQTT, emAssinaturas, escolhaPublicar em um tópico.

   b. NoPublicar, insira o nome do tópico de entrada,**`device/33/data`**.

   c. Copie os dados de amostra mostrados aqui e, na caixa de edição abaixo do nome do tópico, cole os dados de amostra.

```
{
  "temperature": 28,
  "humidity": 80,
  "barometer": 1013,
  "wind": {
    "velocity": 22,
    "bearing": 255
  }
}
```

   d. Para enviar sua mensagem MQTT, escolhaPublicar em um tópico.

A mensagem que enviou na**`device/+/data`**subscrição; no entanto, como o valor da temperatura está abaixo da temperatura máxima na instrução de consulta de regra, você não deve receber uma mensagem de texto.

Se não vir o comportamento correto, verifique as dicas de solução de problemas.

Solução de problemas doAWS LambdaRegra e notificação

Aqui estão algumas coisas para verificar, caso você não esteja vendo os resultados esperados.

- Você recebeu um banner de erro

  Se um erro aparecer quando você publicou a mensagem de entrada, corrija esse erro primeiro. As etapas a seguir podem ajudá-lo a corrigir esse erro.

- A mensagem de entrada não será exibida no MQTT cliente do

  Toda vez que você publica sua mensagem de entrada no`device/32/data`tópico, essa mensagem deve aparecer no cliente MQTT, se você se inscreveu no`device/+/data`Filtro de tópicos como descrito no procedimento.

  ### Coisas para verificar

  - Verifique o filtro de tópicos no qual você se inscreveu

    Se você se inscreveu no tópico da mensagem de entrada conforme descrito no procedimento, verá uma cópia da mensagem de entrada toda vez que a publicar.

    Se você não vir a mensagem, verifique o nome do tópico que você se inscreveu e compare-o com o tópico no qual você publicou. Os nomes de tópicos diferenciam maiúsculas de minúsculas e o tópico no qual você se inscreveu deve ser idêntico ao tópico ao qual você publicou a carga útil da mensagem.

  - Verifique a função de publicação de mensagens

    No Cliente MQTT, emAssinaturas, escolhadispositivo/+/dados, verifique o tópico da mensagem de publicação e, em seguida, escolhaPublicar em um tópico. Você deve ver a carga útil da mensagem na caixa de edição abaixo do tópico aparecer na lista de mensagens.

- Você não recebe uma mensagem SMS

  Para que sua regra funcione, ela deve ter a política correta que a autoriza a receber uma mensagem e enviar uma notificação do SNS, e ela deve receber a mensagem.

  ### Coisas para verificar

  - Verifique oRegião da AWSdo seu cliente MQTT e da regra que você criou

    O console no qual você está executando o cliente MQTT deve estar no mesmoAWSRegião como a regra que você criou.

  - Verifique se o valor da temperatura na carga útil da mensagem excede o limite de teste

    Se o valor da temperatura for menor ou igual a 30, conforme definido na instrução de consulta de regra, a regra não executará nenhuma de suas ações.

  - Verifique o tópico da mensagem de entrada na instrução de consulta de regra

    Para que a regra funcione, ela deve receber uma mensagem com o nome do tópico que corresponde ao filtro de tópico na cláusula FROM da instrução de consulta de regra.

    Verifique a ortografia do filtro de tópico na instrução de consulta de regra com a do tópico no cliente MQTT. Os nomes de tópicos diferenciam maiúsculas e minúsculas e o tópico da mensagem deve corresponder ao filtro de tópico na instrução de consulta de regra

  - Verifique o conteúdo da carga da mensagem de entrada

    Para que a regra funcione, ela deve encontrar o campo de dados na carga da mensagem declarada na instrução SELECT.

Verifique a ortografia do`temperature`campo na instrução de consulta de regra com a da carga útil da mensagem no cliente MQTT. Os nomes dos campos diferenciam letras maiúsculas de`temperature`No comando de consulta de regra deve ser idêntico ao`temperature`campo na carga útil da mensagem.

Certifique-se de que o documento JSON na carga da mensagem esteja formatado corretamente. Se o JSON tiver algum erro, como uma vírgula ausente, a regra não poderá lê-lo.

- Verifique a notificação do Amazon SNS

  DentroEtapa 1: Criar um tópico do Amazon SNS que envia uma mensagem de texto SMS (p. 202), consulte a etapa 3 que descreve como testar a notificação do Amazon SNS e testar a notificação para garantir que a notificação funcione.
- Verifique a função do Lambda

  DentroEtapa 1: Criar umAWS LambdaFunção que envia mensagens de texto (p. 215), consulte a etapa 5 que descreve como testar a função do Lambda usando dados de teste e testar a função do Lambda.
- Verifique a função que está sendo usada pela regra

  A ação da regra deve ter permissão para receber o tópico original e publicar o novo tópico.

  As políticas que autorizam a regra a receber dados de mensagens e republicá-los são específicas dos tópicos usados. Se você alterar o tópico usado para republicar os dados da mensagem, deverá atualizar a função da ação da regra para atualizar sua política para corresponder ao tópico atual.

  Se você suspeitar que esse é o problema, edite a ação Republicar regra e crie uma nova função. Novas funções criadas pela ação de regra recebem as autorizações necessárias para executar essas ações.

Etapa 4: Revise os resultados e as próximas etapas

Neste tutorial:

- Você criou umAWS IoTregra para chamar uma função do Lambda que enviou uma notificação do Amazon SNS que usou sua carga de mensagem personalizada.
- Você usou uma consulta e funções SQL simples em uma instrução de consulta de regra para criar uma nova carga de mensagem para sua função do Lambda.
- Você usou o cliente MQTT do para testar oAWS IoTregra.

Next steps (Próximas etapas)

Depois de enviar algumas mensagens de texto com essa regra, tente experimentar com ela para ver como a mudança de alguns aspectos do tutorial afeta a mensagem e quando ela é enviada. Veja aqui algumas ideias para começar a usar.

- Altere o`device_id`no tópico da mensagem de entrada e observe o efeito no conteúdo da mensagem de texto.
- Altere os campos selecionados na instrução de consulta de regra, atualize a função do Lambda para usá-los em uma nova mensagem e observe o efeito no conteúdo da mensagem de texto.
- Altere o teste na instrução de consulta de regra para testar uma temperatura mínima em vez de uma temperatura máxima. Atualize a função do Lambda para formatar uma nova mensagem e lembre-se de alterar o nome de`max_temperature`.
- Para saber mais sobre como encontrar erros que podem ocorrer enquanto você está desenvolvendo e usandoAWS IoTregras, consulteMonitorar o AWS IoT (p. 432).

Mantendo o estado do dispositivo enquanto o dispositivo está offline com Sombras de dispositivos

Esses tutoriais mostram como usar oAWS IoTServiço Device Shadow para armazenar e atualizar as informações de estado de um dispositivo. O documento Shadow, que é um documento JSON, mostra a alteração no estado do dispositivo com base nas mensagens publicadas por um dispositivo, aplicativo local ou serviço. Neste tutorial, o documento Shadow mostra a mudança na cor de uma lâmpada. Esses tutoriais também mostram como a sombra armazena essas informações mesmo quando o dispositivo está desconectado da Internet e passa as informações de estado mais recentes de volta para o dispositivo quando ele volta on-line e solicita essas informações.

Recomendamos que você experimente esses tutoriais na ordem em que são mostrados aqui, começando com oAWS IoTrecursos que você precisa criar e a configuração de hardware necessária, o que também ajuda a aprender os conceitos de forma incremental. Esses tutoriais mostram como configurar e conectar um dispositivo Raspberry Pi para uso comAWS IoT. Se você não tiver o hardware necessário, você pode seguir esses tutoriais adaptando-os a um dispositivo de sua escolha ou porCriando um dispositivo virtual com o Amazon EC2 (p. 43).

Visão geral do cenário do tutorial

O cenário desses tutoriais é um aplicativo ou serviço local que altera a cor de uma lâmpada e que publica seus dados em tópicos de sombra reservados. Esses tutoriais são semelhantes à funcionalidade Device Shadow descrita natutorial Conceitos básicos (p. 21)e são implementados em um dispositivo Raspberry Pi. Os tutoriais desta seção se concentram em uma única sombra clássica enquanto mostram como você pode acomodar sombras nomeadas ou vários dispositivos.

Os tutoriais a seguir o ajudarão a aprender a usar oAWS IoTServiço Device Shadow.

- Tutorial: Preparando o Raspberry Pi para executar o aplicativo shadow (p. 225)

  Este tutorial mostra como configurar um dispositivo Raspberry Pi para conexão comAWS IoT. Você também criará umAWS IoTdocumento de política e um recurso de coisa, baixe os certificados e, em seguida, anexe a política a esse recurso. A conclusão deste tutorial requer cerca de 30 minutos.
- Tutorial: Instalando o Device SDK e executando o aplicativo de amostra para Device Shadows (p. 229)

  Este tutorial mostra como instalar as ferramentas, o software e o software necessáriosAWS IoTSDK do dispositivo para Python e, em seguida, execute o aplicativo de sombra de amostra. Este tutorial baseia-se nos conceitos apresentados noConnect um Raspberry Pi ou outro dispositivo (p. 57)E leva 20 minutos para ser concluído.
- Tutorial: Interagindo com Device Shadow usando o aplicativo de amostra e o cliente de teste MQTT (p. 236)

  Este tutorial mostra como você usa `oshadow.py`Exemplo de aplicativo eAWS IoTconsoleobservar a interação entreAWS IoTSombras do dispositivo e as mudanças de estado da lâmpada. O tutorial também mostra como enviar mensagens MQTT para os tópicos reservados do Device Shadow. Este tutorial pode demorar 45 minutos para ser concluído.

AWS IoTVisão geral Device Shadow

Um Device Shadow é uma representação virtual persistente de um dispositivo gerenciado por umrecurso de coisas (p. 265)você cria noAWS IoTRegistro. O documento Shadow é um JSON ou umJavaScriptDocumento de notação usado para armazenar e recuperar as informações sobre o estado atual de um dispositivo. Você pode usar a shadow para obter e definir o estado de um dispositivo em tópicos MQTT ou APIs REST HTTP, independentemente de o dispositivo estar conectado à Internet.

Um documento Shadow contém um`state`propriedade que descreve esses aspectos do estado do dispositivo.

- `desired`: Os aplicativos especificam os estados desejados das propriedades do dispositivo atualizando o`desired`objeto.
- `reported`: Os dispositivos relatam seu estado atual no`reported`objeto.
- `delta`:AWS IoTO relata as diferenças entre o estado desejado e relatado no`delta`objeto.

Veja um exemplo de um documento de estado Shadow (Shadow).

```
{
  "state": {
    "desired": {
      "color": "green"
      },
    "reported": {
      "color": "blue"
      },
    "delta": {
      "color": "green"
      }
   }
}
```

Para atualizar o documento Shadow de um dispositivo, você pode usar oTópicos MQTT reservados (p. 114), oREST APIs Device Shadow (p. 659)que suportam o`GET`,`UPDATE`, e`DELETE`operações com HTTP e oAWS IoTCLI.

No exemplo anterior, digamos que você queira mudar o`desired`Cor para`yellow`. Para fazer isso, envie uma solicitação aoUpdateThingShadow (p. 661)API ou publique uma mensagem naAtualização (p. 666)tópico,`$aws/things/THING_NAME/shadow/update`.

```
{
  "state": {
    "desired": {
      "color": yellow
    }
  }
}
```

As atualizações afetam apenas os campos especificados na solicitação. Depois de atualizar com êxito o Device Shadow,AWS IoTPublica o novo`desired`estado para o`delta`tópico,`$aws/things/THING_NAME/shadow/delta`. O documento Shadow neste caso se parece com isso:

```
{
  "state": {
    "desired": {
      "color": yellow
    },
    "reported": {
      "color": green
    },
    "delta": {
      "color": yellow
      }
  }
}
```

O novo estado é então relatado aoAWS IoTDevice Shadow usando o`Update`tópico`$aws/things/THING_NAME/shadow/update`Com a seguinte mensagem JSON:

```
{
```

```
  "state": {
    "reported": {
      "color": yellow
    }
  }
}
```

Se você quiser obter as informações sobre o estado atual, envie uma solicitação para oGetThingShadow (p. 660)API ou publique uma mensagem MQTT noComece (p. 664)tópico,`$aws/things/THING_NAME/shadow/get`.

Para obter mais informações sobre como usar o serviço Device Shadow, consulteServiço Device Shadow do AWS IoT (p. 633).

Para obter mais informações sobre como usar Device Shadows em dispositivos, aplicativos e serviços, consulteUsar sombras em dispositivos (p. 637)eUsar sombras em aplicativos e serviços (p. 640).

Para obter informações sobre como interagir comAWS IoTsombras, consulteInteragir com sombras (p. 652).

Para obter informações sobre tópicos reservados MQTT e APIs REST HTTP, consulteTópicos MQTT do Device Shadow (p. 663)eAPI REST do Device Shadow (p. 659).

Tutorial: Preparando o Raspberry Pi para executar o aplicativo shadow

Este tutorial demonstra como configurar e configurar um dispositivo Raspberry Pi e criar oAWS IoTrecursos que um dispositivo requer para conectar e trocar mensagens MQTT.

> **Note**
>
> Se você está planejandothe section called “Criar um dispositivo virtual com o Amazon EC2” (p. 43), você pode pular esta página e continuar athe section called “Configurar o dispositivo” (p. 43). Você criará esses recursos ao criar sua coisa virtual. Se você quiser usar um dispositivo diferente em vez do Raspberry Pi, você pode tentar seguir esses tutoriais adaptando-os a um dispositivo de sua escolha.

Neste tutorial, você aprenderá a:

- Configure um dispositivo Raspberry Pi e configure-o para uso comAWS IoT.
- Criar umAWS IoTdocumento de política, que autoriza seu dispositivo a interagir comAWS IoTServiços da .
- Criar um recurso de coisa noAWS IoTos certificados de dispositivo X.509 e, em seguida, anexe o documento de política.

  O problema é a representação virtual do seu dispositivo naAWS IoTRegistro. O certificado autentica seu dispositivo paraAWS IoTCore, e o documento de política autoriza seu dispositivo a interagir comAWS IoT.

Como executar este tutorial

Para executar a`shadow.py`aplicativo de amostra para Device Shadows, você precisará de um dispositivo Raspberry Pi que se conecte aAWS IoT. Recomendamos que você siga este tutorial na ordem em que é apresentado aqui, começando com a configuração do Raspberry Pi e seus acessórios, e depois criando uma política e anexando a política a um recurso que você cria. Você pode seguir este tutorial usando a interface gráfica do usuário (GUI) suportada pelo Raspberry Pi para abrir oAWS IoTconsole no

navegador da web do dispositivo, o que também facilita o download dos certificados diretamente para o seu Raspberry Pi para conexão comAWS IoT.

Antes de começar este tutorial, verifique se você tem:

- Uma Conta da AWS. Se você ainda não tiver um, conclua as etapas descritas emConfigurar seuConta da AWS (p. 19)Antes de continuar. Você precisará doConta da AWSeAWS IoTConsole para concluir este tutorial.
- O Raspberry Pi e seus acessórios necessários. Você precisará de:
  - UMARaspberry Pi 3 Modelo Bou modelo mais recente. Este tutorial pode funcionar em versões anteriores do Raspberry Pi, mas não o testamos.
  - SO Raspberry Pi (32 bits)ou posterior. Recomendamos o uso da versão mais recente do Raspberry Pi OS. Versões anteriores do sistema operacional podem funcionar, mas não o testamos.
  - Uma conexão Ethernet ou Wi-Fi.
  - Teclado, mouse, monitor, cabos e fontes de alimentação.

A conclusão deste tutorial requer cerca de 30 minutos.

Etapa 1: Configurar e configurar o dispositivo Raspberry Pi

Nesta seção, configuraremos um dispositivo Raspberry Pi para uso comAWS IoT.

> **Important**
>
> Adaptar essas instruções a outros dispositivos e sistemas operacionais pode ser um desafio. Você precisará entender o dispositivo bem o suficiente para poder interpretar essas instruções e aplicá-las ao seu dispositivo. Se você encontrar dificuldades, você pode tentar uma das outras opções de dispositivo como alternativa, comoCriar um dispositivo virtual com o Amazon EC2 (p. 43)ouUse seu PC ou Mac Windows ou Linux como umAWS IoTdispositivo (p. 51).

Você precisará configurar seu Raspberry Pi para que ele possa iniciar o sistema operacional (SO), conectar-se à Internet e permitir que você interaja com ele em uma interface de linha de comando. Você também pode usar a interface gráfica de usuário (GUI) suportada com o Raspberry Pi para abrir oAWS IoTConsole e execute o restante deste tutorial.

Para configurar o Raspberry Pi

1. Insira o cartão SD no slot para cartão microSD no Raspberry Pi. Alguns cartões SD vêm pré-carregados com um gerenciador de instalação que solicita um menu para instalar o sistema operacional depois de inicializar a placa. Você também pode usar o imager Raspberry Pi para instalar o sistema operacional em seu cartão.
2. Connect uma TV ou monitor HDMI ao cabo HDMI que se conecta à porta HDMI do Raspberry Pi.
3. Connect o teclado e o mouse às portas USB do Raspberry Pi e, em seguida, conecte o adaptador de energia para inicializar a placa.

Após a inicialização do Raspberry Pi, se o cartão SD vier pré-carregado com o gerenciador de instalação, um menu aparecerá para instalar o sistema operacional. Se você tiver problemas para instalar o sistema operacional, tente as etapas a seguir. Para obter mais informações sobre a configuração do Raspberry Pi, consulteConfigurar seu Raspberry Pi.

Se estiver tendo problemas para configurar o Raspberry Pi:

- Verifique se você inseriu o cartão SD antes de inicializar a placa. Se você conectar o cartão SD depois de inicializar a placa, o menu de instalação pode não aparecer.

- Certifique-se de que a TV ou o monitor esteja ligado e a entrada correta esteja selecionada.
- Certifique-se de que você está usando o software compatível com Raspberry Pi.

Depois de instalar e configurar o sistema operacional Raspberry Pi, abra o navegador da Web do Raspberry Pi e navegue até oAWS IoT CoreConsole para continuar o restante das etapas deste tutorial.

Se você puder abrir oAWS IoT Coreconsole, você está Raspberry Pi está pronto e você pode continuarthe section called “Provisionando seu dispositivo emAWS IoT ” (p. 227).

Se estiver tendo problemas ou precisar de ajuda adicional, consulteReceber ajuda para o Raspberry Pi.

Tutorial: Provisionando seu dispositivo emAWS IoT

Esta seção cria oAWS IoT Corerecursos que seu tutorial usará.

Etapas para provisionar seu dispositivo emAWS IoT

- Etapa 1: Criar umAWS IoTPolítica da Device Shadow (p. 227)
- Etapa 2: Crie um recurso de coisa e anexe a política à coisa (p. 228)
- Etapa 3: Analise os resultados e as próximas etapas (p. 229)

Etapa 1: Criar umAWS IoTPolítica da Device Shadow

Os certificados X.509 autenticam seu dispositivo comAWS IoT Core.AWS IoTas políticas são anexadas ao certificado que permite que o dispositivo executeAWS IoTAs operações, como se inscrever em tópicos reservados MQTT ou publicar neles usados pelo serviço Device Shadow (Device Shadow). Seu dispositivo apresenta o certificado ao se conectar e enviar mensagens paraAWS IoT Core.

Neste procedimento, você criará uma política do que conceda ao dispositivo executar oAWS IoTOperações necessárias para executar o programa de exemplo. Recomendamos que você crie uma política que conceda apenas as permissões necessárias para executar a tarefa. Você cria oAWS IoTdiretiva primeiro e, em seguida, anexe-a ao certificado do dispositivo que você criará posteriormente.

Para criar uma política do AWS IoT

1. No menu esquerdo, escolhaSecuree, depois, escolhaPolíticas. Se sua conta tiver políticas existentes, escolhaCriar, caso contrário, noVocê ainda não tem uma políticaPágina, escolhaCriar uma política.
2. Na página Create a policy (Criar uma política):

   a. Insira um nome para a política noName (Nome)campo (por exemplo,**`My_Device_Shadow_policy`**). Não use informações de identificação pessoal nos nomes de política.

   b. No documento de política, você descreve ações de conexão, assinatura, recebimento e publicação que dão ao dispositivo permissão para publicar e assinar os tópicos reservados do MQTT.

      Copie a política de exemplo a seguir e cole-a no documento de política. Substituir`thingname`com o nome da coisa que você criará (por exemplo,`My_light_bulb`),`region`com oAWS IoTRegião onde você está usando os serviços e`account`com suas instruçõesConta da AWSnúmero. Para obter mais informações sobreAWS IoTPolíticas, consultePolíticas do AWS IoT Core (p. 332).

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

```
      "Action": [
        "iot:Publish"
      ],
      "Resource": [
        "arn:aws:iot:region:account:topic/$aws/things/thingname/shadow/get",
        "arn:aws:iot:region:account:topic/$aws/things/thingname/shadow/update"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "iot:Receive"
      ],
      "Resource": [
        "arn:aws:iot:region:account:topic/$aws/things/thingname/shadow/get/
accepted",
        "arn:aws:iot:region:account:topic/$aws/things/thingname/shadow/get/
rejected",
        "arn:aws:iot:region:account:topic/$aws/things/thingname/shadow/update/
accepted",
        "arn:aws:iot:region:account:topic/$aws/things/thingname/shadow/update/
rejected",
        "arn:aws:iot:region:account:topic/$aws/things/thingname/shadow/update/
delta"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "iot:Subscribe"
      ],
      "Resource": [
        "arn:aws:iot:region:account:topicfilter/$aws/things/thingname/shadow/get/
accepted",
        "arn:aws:iot:region:account:topicfilter/$aws/things/thingname/shadow/get/
rejected",
        "arn:aws:iot:region:account:topicfilter/$aws/things/thingname/shadow/
update/accepted",
        "arn:aws:iot:region:account:topicfilter/$aws/things/thingname/shadow/
update/rejected",
        "arn:aws:iot:region:account:topicfilter/$aws/things/thingname/shadow/
update/delta"
      ]
    },
    {
      "Effect": "Allow",
      "Action": "iot:Connect",
      "Resource": "arn:aws:iot:region:account:client/test-*"
    }
  ]
}
```

Etapa 2: Crie um recurso de coisa e anexe a política à coisa

Dispositivos conectados aoAWS IoTpode ser representado porRecursos de coisasnoAWS IoTRegistro. UMArecurso de coisasRepresenta um dispositivo específico ou uma entidade lógica, como a lâmpada neste tutorial.

Para saber como criar uma coisa emAWS IoT, siga as etapas descritas emCriar um objeto de coisa (p. 41). Aqui estão algumas coisas importantes a serem observadas ao seguir as etapas desse tutorial:

1. SelecioneCriar uma única coisa, e noName (Nome), insira um nome para a coisa que seja a mesma `dothingname`(por exemplo:`My_light_bulb`) especificou quando criou a política anteriormente.

Não é possível alterar o nome de uma coisa após sua criação. Se você lhe deu um nome diferente que não seja`thingname`, crie uma coisa nova com o nome como`thingname`e exclua a coisa antiga.

> **Note**
>
> Não use informações de identificação pessoal no nome da coisa. O nome da coisa pode aparecer em comunicações e relatórios não criptografados.

2. Recomendamos que você baixe cada um dos arquivos de certificado noCertificado criado!em um local onde você pode encontrá-los facilmente. Você precisará instalar esses arquivos para executar o aplicativo de amostra.

   Recomendamos que você faça download dos arquivos em um`certs`subdiretório em seu`home`no Raspberry Pi e nomeie cada um deles com um nome mais simples, conforme sugerido na tabela a seguir.

   Nomes de arquivos de certificado

Arquivo	Caminho do arquivo
Certificado CA raiz	`~/certs/Amazon-root-CA-1.pem`
Certificado de dispositivo	`~/certs/device.pem.crt`
Chave privada	`~/certs/private.pem.key`

3. Depois de ativar o certificado para habilitar conexões comAWS IoT, escolhaAnexar uma políticaE certifique-se de anexar a política criada anteriormente (por exemplo,**`My_Device_Shadow_policy`**) para a coisa.

   Depois de criar uma coisa, você pode ver seu recurso de coisa exibido na lista de coisas naAWS IoTconsole do .

Etapa 3: Analise os resultados e as próximas etapas

Neste tutorial, você aprendeu a:

- Configure e configure o dispositivo Raspberry Pi.
- Criar umAWS IoTdocumento de política que autoriza seu dispositivo a interagir comAWS IoTServiços da .
- Crie um recurso de coisa e um certificado de dispositivo X.509 associado e anexe o documento de política a ele.

Próximas etapas

Agora é possível instalar oAWS IoTSDK de dispositivo para Python, execute `oshadow.py`aplicativo de amostra e use Sombras de dispositivos para controlar o estado. Para obter mais informações sobre como executar este tutorial, consulteTutorial: Instalando o Device SDK e executando o aplicativo de amostra para Device Shadows (p. 229).

Tutorial: Instalando o Device SDK e executando o aplicativo de amostra para Device Shadows

Esta seção mostra como é possível instalar o software necessário e oAWS IoTSDK de dispositivo para Python e execute `oshadow.py`aplicativo de amostra para editar o documento Shadow e controlar o estado da sombra.

Neste tutorial, você aprenderá a:

- Use o software instalado eAWS IoTSDK do dispositivo para Python para executar o aplicativo de amostra.
- Saiba como inserir um valor usando o aplicativo de amostra publica o valor desejado naAWS IoTconsole do .
- Analise `oshadow.py`aplicativo de amostra e como ele usa o protocolo MQTT para atualizar o estado da sombra.

Antes de executar este tutorial:

Você deve ter configurado oConta da AWS, configurou seu dispositivo Raspberry Pi e criou umAWS IoTcoisa e política que dão ao dispositivo permissões para publicar e assinar os tópicos reservados do MQTT do serviço Device Shadow. Para obter mais informações, consulte Tutorial: Preparando o Raspberry Pi para executar o aplicativo shadow (p. 225).

Você também deve ter instalado o Git, o Python e oAWS IoTSDK de dispositivo para Python. Este tutorial se baseia nos conceitos apresentados no tutorialConnect um Raspberry Pi ou outro dispositivo (p. 57). Se você ainda não tentou esse tutorial, recomendamos que siga as etapas descritas nesse tutorial para instalar os arquivos de certificado e o Device SDK e, em seguida, volte a este tutorial para executar `oshadow.py`App de amostra.

Neste tutorial, você irá:

- Etapa 1: Execute o aplicativo de amostra shadow.py (p. 230)
- Etapa 2: Revise o aplicativo de exemplo do SDK do dispositivo shadow.py (p. 233)
- Etapa 3: Solução de problemas com oshadow.pyApp de amostra (p. 234)
- Etapa 4: Analise os resultados e as próximas etapas (p. 235)

A conclusão deste tutorial requer cerca de 20 minutos.

Etapa 1: Execute o aplicativo de amostra shadow.py

Antes de executar `oshadow.py`aplicativo de exemplo, você precisará das seguintes informações, além dos nomes e localização dos arquivos de certificado que você instalou.

Valores de parâmetros do aplicativo

Parâmetro	Onde encontrar o valor
`your-iot-thing-Name`	Nome doAWS IoTcoisa que você criou anteriormentethe section called “Etapa 2: Crie um recurso de coisa e anexe a política à coisa” (p. 228). Para encontrar esse valor, noAWS IoTconsole, escolhaGerenciee, depois, escolhaCoisas.
`your-iot-endpoint`	O*`your-iot-endpoint`*value tem um formato de:*`endpoint_id`*-`ats.iot.`*`region`*`.amazonaws.com`, por exemplo,`a3qj468EXAMPLE-ats.iot.us-west-2.amazonaws.com`. Para localizar esse valor: 1. NoAWS IoTconsole, escolhaGerenciee, depois, escolhaCoisas.

Parâmetro	Onde encontrar o valor
	2. Escolha a coisa de IoT que você criou para o seu dispositivo,my_light_bulb, que você usou anteriormente e depois escolhaInteragir. Na página de detalhes da coisa, o endpoint é exibido naHTTPSseção.

Instale e execute o aplicativo de amostra

1. Navegue até o diretório de aplicativo de exemplo.

```
cd ~/aws-iot-device-sdk-python-v2/samples
```

2. Na janela da linha de comando, substitua*your-iot-endpoint*e*your-iot-thing-Name*conforme indicado e execute este comando.

```
python3 shadow.py --ca_file ~/certs/Amazon-root-CA-1.pem --cert ~/certs/device.pem.crt
 --key ~/certs/private.pem.key --endpoint your-iot-endpoint --thing_name your-iot-
thing-name
```

3. Observe que o aplicativo de amostra:

   1. Conecta-se aoAWSServiço de IoT para sua conta.
   2. Subscreve no`Delta`eventos e`Update`e`Get`respostas.
   3. Solicita que você insira um valor desejado no terminal.
   4. Exibe uma saída semelhante à seguinte:

```
Connecting to a3qEXAMPLEffp-ats.iot.us-west-2.amazonaws.com with client ID
 'test-0c8ae2ff-cc87-49d2-a82a-ae7ba1d0ca5a'...
Connected!
Subscribing to Delta events...
Subscribing to Update responses...
Subscribing to Get responses...
Requesting current shadow state...
Launching thread to read user input...
Finished getting initial shadow state.
Shadow contains reported value 'off'.
Enter desired value:
```

> **Note**
>
> Se estiver tendo problemas para executar o`shadow.py`Exemplo de aplicativo, revisãothe section called “Etapa 3: Solução de problemas com o`shadow.py`App de amostra” (p. 234). Para obter informações adicionais que possam ajudá-lo a corrigir o problema, adicione o`--verbosity debug`parâmetro para a linha de comando para que o aplicativo de amostra exiba mensagens detalhadas sobre o que ele está fazendo.

Insira valores e observe as atualizações no documento Shadow

Você pode inserir valores no terminal para especificar o`desired`value, que também atualiza o`reported`value. Digamos que você insira a cor`yellow`No terminal. O`reported`valor também é atualizado para a cor`yellow`. O seguinte mostra as mensagens exibidas no terminal:

```
Enter desired value:
```

```
yellow
Changed local shadow value to 'yellow'.
Updating reported shadow value to 'yellow'...
Update request published.
Finished updating reported shadow value to 'yellow'.
```

Quando você publica essa solicitação de atualização,AWS IoTcria uma sombra clássica padrão para o recurso thing. Você pode observar a solicitação de atualização que você publicou no`reportededesired`Valores noAWS IoTconsole observando o documento Shadow para o recurso coisa que você criou (por exemplo,`My_light_bulb`). Para ver a atualização no documento Shadow:

1. NoAWS IoTConsole, escolhaGerenciee, depois, escolhaCoisas.
2. Na lista de coisas exibidas, selecione a coisa que você criou, escolhaSombrase, depois, escolhaClassic Shadow.

O documento Shadow deve ser semelhante ao seguinte:`reportededesired`valores definidos para a cor`yellow`. Você vê esses valores noEstado de sombraseção do documento.

```
{
"desired": {
  "welcome": "aws-iot",
  "color": "yellow"
},
"reported": {
  "welcome": "aws-iot",
  "color": "yellow"
}
}
```

Você também vê umMetadadosque contém as informações de carimbo de data/hora e o número da versão da solicitação.

Você pode usar a versão do documento de estado para garantir que está atualizando a versão mais recente do documento Shadow de um dispositivo. Se você enviar outra solicitação de atualização, o número da versão aumentará em 1. Ao fornecer uma versão com uma solicitação de atualização, o serviço rejeitará a solicitação com um código de resposta de conflito HTTP 409 se a versão atual do documento de estado não corresponder à versão fornecida.

```
{
"metadata": {
  "desired": {
    "welcome": {
      "timestamp": 1620156892
    },
    "color": {
      "timestamp": 1620156893
    }
  },
  "reported": {
    "welcome": {
      "timestamp": 1620156892
    },
    "color": {
      "timestamp": 1620156893
    }
  }
},
"version": 10
}
```

Para saber mais sobre o documento Shadow e observar as alterações nas informações de estado, prossiga para o próximo tutorialTutorial: Interagindo com Device Shadow usando o aplicativo de amostra e o cliente de teste MQTT (p. 236)Como descrito noEtapa 4: Analise os resultados e as próximas etapas (p. 235)Seção deste tutorial. Como opção, você também pode aprender sobre `oshadow.py`exemplo de código e como ele usa o protocolo MQTT na seção a seguir.

Etapa 2: Revise o aplicativo de exemplo do SDK do dispositivo shadow.py

Esta seção analisa `oshadow.py`Exemplo de aplicativo doAWS IoTSDK de dispositivo v2 para PythonUsado neste tutorial. Aqui, analisaremos como ele se conecta aoAWS IoT Coreusando o protocolo MQTT e MQTT sobre WSS. OAWSTempo de execução comum (AWS-CRT)biblioteca fornece suporte a protocolos de comunicação de baixo nível e está incluída noAWS IoTSDK de dispositivo v2 para Python.

Enquanto este tutorial usa MQTT e MQTT sobre WSS,AWS IoToferece suporte a dispositivos que publicam solicitações HTTPS. Para obter um exemplo de programa Python que envia uma mensagem HTTP de um dispositivo, consulte aExemplo de código HTTPS (p. 97)Uso do Python's`requests`biblioteca.

Para obter informações sobre como você pode tomar uma decisão informada sobre qual protocolo usar para as comunicações do seu dispositivo, consulte oEscolhendo um protocolo para a comunicação do seu dispositivo (p. 85).

MQTT

O`shadow.py`Chamadas de exemplo`mtls_from_path`(mostrado aqui) no`mqtt_connection_builder`Para estabelecer uma conexão comAWS IoT CoreUsando o protocolo MQTT.`mtls_from_path`usa certificados X.509 e TLS v1.2 para autenticar o dispositivo. OAWS-A biblioteca CRT lida com os detalhes de nível inferior dessa conexão.

```
mqtt_connection = mqtt_connection_builder.mtls_from_path(
  endpoint=args.endpoint,
  cert_filepath=args.cert,
  pri_key_filepath=args.key,
  ca_filepath=args.ca_file,
  client_bootstrap=client_bootstrap,
  on_connection_interrupted=on_connection_interrupted,
  on_connection_resumed=on_connection_resumed,
  client_id=args.client_id,
  clean_session=False,
  keep_alive_secs=6
)
```

- `endpoint`é seuAWS IoTendpoint que você passou da linha de comando e`client_id`é o ID que identifica exclusivamente este dispositivo naRegião da AWS.
- `cert_filepath`,`pri_key_filepath`, e`ca_filepath`são os caminhos para o certificado e os arquivos de chave privada do dispositivo, e o arquivo CA raiz.
- `client_bootstrap`é o objeto de tempo de execução comum que lida com atividades de comunicação de soquete e é instanciado antes da chamada para`mqtt_connection_builder.mtls_from_path`.
- `on_connection_interrupted`e`on_connection_resumed`são funções de retorno de chamada a serem chamadas quando a conexão do dispositivo é interrompida e retomada.
- `clean_session`é se deseja iniciar uma sessão nova e persistente ou, se houver uma, reconecte-se a uma existente.`keep_alive_secs`é o valor de manter vivo, em segundos, para enviar`CONNECT`solicitação. Um ping será enviado automaticamente nesse intervalo. O servidor pressupõe que a conexão seja perdida se não receber um ping após 1,5 vezes esse valor.

O`shadow.py`amostra também chama`websockets_with_default_aws_signing`no`mqtt_connection_builder`Para estabelecer uma conexão comAWS IoT Coreusando o protocolo MQTT sobre WSS. O MQTT over WSS também usa os mesmos parâmetros que o MQTT e usa estes parâmetros adicionais:

- `regione` oAWSassinatura Região usada pela autenticação Signature V4 `ecredentials_provider`é oAWScredenciais fornecidas para uso para autenticação. A região é passada da linha de comando e `ocredentials_provider`objeto é instanciado pouco antes da chamada para`mqtt_connection_builder.websockets_with_default_aws_signing`.
- `websocket_proxy_options`são as opções de proxy HTTP, se estiver usando um host proxy. No`shadow.py`aplicativo de amostra, esse valor é instanciado pouco antes da chamada para`mqtt_connection_builder.websockets_with_default_aws_signing`.

Assine tópicos e eventos do Shadow

O`shadow.py`O exemplo de tenta estabelecer uma conexão e aguarda para estar totalmente conectada. Se não estiver conectado, os comandos serão enfileirados. Uma vez conectado, a amostra assina eventos delta e atualiza e recebe mensagens, e publica mensagens com um nível de Qualidade de Serviço (QoS) de 1 (`mqtt.QoS.AT_LEAST_ONCE`).

Quando um dispositivo se inscreve em uma mensagem com QoS nível 1, o corretor de mensagens salva as mensagens que o dispositivo está inscrito até que elas possam ser enviadas para o dispositivo. O corretor de mensagens reenvia as mensagens até receber um`PUBACK`resposta do dispositivo.

Para obter mais informações sobre o protocolo MQTT, consulteRevise o protocolo MQTT (p. 178)eMQTT (p. 86).

Para obter mais informações sobre como MQTT, MQTT sobre WSS, sessões persistentes e níveis de QoS usados neste tutorial, consulteRevise o aplicativo de exemplo do SDK do dispositivo pubsub.py (p. 178).

Etapa 3: Solução de problemas com `oshadow.py`App de amostra

Quando você executa `oshadow.py`aplicativo de amostra, você deve ver algumas mensagens exibidas no terminal e um prompt para inserir um`desired`value. Se o programa lançar um erro, então, para depurar o erro, você pode começar verificando se você executou o comando correto para o seu sistema.

Em alguns casos, a mensagem de erro pode indicar problemas de conexão e parecer semelhante a:`Host name was invalid for dns resolution`ou`Connection was closed unexpectedly`. Nesses casos, veja algumas coisas que você pode verificar:

- Verifique o endereço do endpoint no comando

  Analise `oendpoint`argumento no comando que você inseriu para executar o aplicativo de amostra, (por exemplo,`a3qEXAMPLEffp-ats.iot.us-west-2.amazonaws.com`) e verifique esse valor noAWS IoTconsole.

  Para verificar se você usou o valor correto:

  1. NoAWS IoTconsole, escolhaGerenciee, depois, escolhaCoisas.
  2. Escolha o que você criou para o aplicativo de exemplo (por exemplo,my_light_bulb) e, depois, escolhaInteragir.

  Na página de detalhes da coisa, o endpoint é exibido naHTTPSseção. Você também verá uma mensagem que diz:`This thing already appears to be connected.`
- Verifique a ativação do certificado

  Certificados autenticam seu dispositivo comAWS IoT Core.

  Para verificar se o certificado está ativo:

  1. NoAWS IoTconsole, escolhaGerenciee, depois, escolhaCoisas.
  2. Escolha o que você criou para o aplicativo de exemplo (por exemplo,my_light_bulb) e, depois, escolhaSegurança.

3. Selecione o certificado e, na página de detalhes do certificado, escolha Selecionar o certificado e, na página de detalhes do certificado, escolhaAções.

Se estiver na lista suspensaAtivarnão está disponível e você só pode escolherDesativar, seu certificado está ativo. Caso contrário, escolhaAtivarE execute novamente o programa de amostra.

Se o programa ainda não for executado, verifique os nomes dos arquivos de certificado na`certs`folder.

- Verifique a política anexada ao recurso coisa

  Enquanto os certificados autenticam seu dispositivo,AWS IoTpolíticas permitem que o dispositivo executeAWS IoTAs operações, como se inscrever em tópicos reservados ao MQTT ou publicar neles.

  Para verificar se a política correta está anexada:

  1. Encontre o certificado conforme descrito anteriormente e, em seguida, escolhaPolíticas.
  2. Escolha a política exibida e verifique se ela descreve o`connect`,`subscribe`,`receive`, e`publish`ações que dão ao dispositivo permissão para publicar e assinar os tópicos reservados do MQTT.

     Para obter um exemplo de política, consulteEtapa 1: Criar umAWS IoTPolítica da Device Shadow (p. 227).

  Se você vir mensagens de erro que indicam problemas para se conectar aoAWS IoT, pode ser por causa das permissões que você está usando para a política. Se for esse o caso, recomendamos começar com uma política que conceda acesso total aoAWS IoTRecursos e, em seguida, execute novamente o programa de exemplo. Você pode editar a política atual ou escolher a política atual, escolherDesconectare, em seguida, crie outra política que forneça acesso total e anexe-a ao seu recurso. Posteriormente, você pode restringir a política apenas às ações e políticas necessárias para executar o programa.

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

- Verifique a instalação do Device SDK

  Se o programa ainda não for executado, você poderá reinstalar o SDK do dispositivo para garantir que a instalação do SDK esteja completa e correta.

Etapa 4: Analise os resultados e as próximas etapas

Neste tutorial, você aprendeu a:

- Instalar o software, as ferramentas e oAWS IoTSDK de dispositivo para Python.
- Entenda como o aplicativo de amostra,`shadow.py`, usa o protocolo MQTT para recuperar e atualizar o estado atual da sombra.
- Execute o aplicativo de amostra para Device Shadows e observe a atualização para o documento Shadow naAWS IoTconsole do . Você também aprendeu a solucionar problemas e corrigir erros ao executar o programa.

Próximas etapas

Agora é possível executar `oshadow.py`aplicativo de amostra e use Sombras de dispositivos para controlar o estado. Você pode observar as atualizações do documento Shadow naAWS IoTConsole e observe eventos delta aos quais o aplicativo de amostra responde. Usando o cliente de teste MQTT, você pode assinar os tópicos de sombra reservados e observar as mensagens recebidas pelos tópicos ao executar o programa de amostra. Para obter mais informações sobre como executar este tutorial, consulteTutorial: Interagindo com Device Shadow usando o aplicativo de amostra e o cliente de teste MQTT (p. 236).

Tutorial: Interagindo com Device Shadow usando o aplicativo de amostra e o cliente de teste MQTT

Para interagir com `oshadow.py`aplicativo de amostra, insira um valor no terminal para `odesired`value. Por exemplo, você pode especificar cores que se assemelham aos semáforos eAWS IoTresponde à solicitação e atualiza os valores relatados.

Neste tutorial, você aprenderá a:

- Usar `ashadow.py`aplicativo de amostra para especificar estados desejados e atualizar o estado atual da sombra.
- Edite o documento Shadow para observar eventos delta e como `oshadow.py`aplicativo de amostra responde a ele.
- Use o cliente de teste MQTT para assinar tópicos de sombra e observar atualizações ao executar o programa de amostra.

Antes de executar este tutorial, você deve ter:

Configurar seuConta da AWS, configurou seu dispositivo Raspberry Pi e criou umAWS IoTcoisa e política. Você também deve ter instalado o software necessário, o Device SDK, os arquivos de certificado e executar o programa de amostra no terminal. Para obter mais informações, consulte os tutoriais anterioresTutorial: Preparando o Raspberry Pi para executar o aplicativo shadow (p. 225)eEtapa 1: Execute o aplicativo de amostra shadow.py (p. 230). Se ainda não tiver feito isso, conclua esses tutoriais.

Neste tutorial, você irá:

- Etapa 1: Atualize os valores desejados e relatados usandoshadow.pyApp de amostra (p. 236)
- Etapa 2: Exibir mensagens doshadow.pyAplicativo de exemplo no cliente de teste MQTT (p. 238)
- Etapa 3: Solucionar problemas com interações Device Shadow (p. 242)
- Etapa 4: Analise os resultados e as próximas etapas (p. 243)

A conclusão deste tutorial requer cerca de 45 minutos.

Etapa 1: Atualize os valores desejados e relatados usando`shadow.py`App de amostra

No tutorial anteriorEtapa 1: Execute o aplicativo de amostra shadow.py (p. 230), você aprendeu a observar uma mensagem publicada no documento Shadow naAWS IoTconsole quando você insere um valor desejado, conforme descrito na seçãoTutorial: Instalando o Device SDK e executando o aplicativo de amostra para Device Shadows (p. 229).

No exemplo anterior, definimos a cor desejada para`yellow`. Depois de inserir cada valor, o terminal solicita que você insira outro`desired`value. Se você inserir novamente o mesmo valor (`yellow`), o aplicativo reconhece isso e solicita que você insira um novo`desired`value.

```
Enter desired value:
yellow
```

```
Local value is already 'yellow'.
Enter desired value:
```

Agora, digamos que você insira a cor`green`.AWS IoTresponde à solicitação e atualiza o`reported`valor para`green`. É assim que a atualização acontece quando o`desired`O estado é diferente do`reported`state, causando um delta.

Como o**`shadow.py`**aplicativo de amostra simula interações Device Shadow:

1. Digite um`desired`valor (digamos`yellow`) no terminal para publicar o estado desejado.
2. Como o`desired`O estado é diferente do`reported`state (digamos a cor`green`), ocorre um delta e o aplicativo que está inscrito no delta recebe essa mensagem.
3. O aplicativo responde à mensagem e atualiza seu estado para o`desired`valor,`yellow`.
4. Em seguida, o aplicativo publica uma mensagem de atualização com o novo valor relatado do estado do dispositivo,`yellow`.

A seguir, mostra as mensagens exibidas no terminal que mostra como a solicitação de atualização é publicada.

```
Enter desired value:
green
Changed local shadow value to 'green'.
Updating reported shadow value to 'green'...
Update request published.
Finished updating reported shadow value to 'green'.
```

NoAWS IoTconsole, o documento Shadow reflete o valor atualizado para`green`para ambos`reported`e`desired`Os campos e o número da versão é incrementado em 1. Por exemplo, se o número da versão anterior for exibido como 10, o número da versão atual será exibido como 11.

> **Note**
>
> A exclusão de uma sombra não redefine o número da versão para 0. Você verá que a versão shadow é incrementada em 1 quando você publica uma solicitação de atualização ou cria outra sombra com o mesmo nome.

Edite o documento Shadow para observar eventos delta

O`shadow.py`aplicativo de amostra também está inscrito`delta`eventos, e responde quando há uma mudança no`desired`value. Por exemplo, você pode alterar o`desired`valor para a cor`red`. Para fazer isso, noAWS IoTconsole, edite o documento Shadow clicando emEditee depois defina o`desired`valor para`red`no JSON, mantendo o`reported`valor para`green`. Antes de salvar as alterações, mantenha o terminal no Raspberry Pi aberto, pois você verá as mensagens exibidas no terminal quando a alteração ocorrer.

```
{
"desired": {
  "welcome": "aws-iot",
  "color": "red"
},
"reported": {
  "welcome": "aws-iot",
  "color": "green"
}
}
```

Depois de salvar o novo valor, o`shadow.py`aplicativo de amostra responde a essa alteração e exibe mensagens no terminal indicando o delta. Em seguida, você deve ver as seguintes mensagens exibidas abaixo do prompt para inserir o`desired`value.

```
Enter desired value:
Received shadow delta event.
Delta reports that desired value is 'red'. Changing local value...
Changed local shadow value to 'red'.
Updating reported shadow value to 'red'...
Finished updating reported shadow value to 'red'.
Enter desired value:
Update request published.
Finished updating reported shadow value to 'red'.
```

Etapa 2: Exibir mensagens do`shadow.py`Aplicativo de exemplo no cliente de teste MQTT

Você pode usar oCliente de teste MQTTnoAWS IoTconsolepara monitorar mensagens MQTT que são passadas em seuConta da AWS. Ao se inscrever em tópicos MQTT reservados usados pelo serviço Device Shadow, você pode observar as mensagens recebidas pelos tópicos ao executar o aplicativo de amostra.

Se você ainda não tiver usado o cliente de teste MQTT, é possível revisarVisualizar mensagens MQTT com o cliente MQTT do AWS IoT (p. 66). Isso ajuda a aprender a usar oCliente de teste MQTTnoAWS IoTconsolePara exibir as mensagens MQTT à medida que elas passam pelo operador de mensagens.

1. Abra o cliente de teste MQTT

   Abrir oCliente de teste MQTT noAWS IoTconsoleem uma nova janela para que você possa observar as mensagens recebidas pelos tópicos do MQTT sem perder a configuração do seu cliente de teste MQTT. O cliente de teste MQTT não retém assinaturas ou registros de mensagens se você deixar ele ir para outra página no console. Para esta seção do tutorial, você pode ter o documento Shadow do seuAWS IoTe o cliente de teste MQTT abrem em janelas separadas para observar mais facilmente a interação com Device Shadows.
2. Assine os tópicos do Shadow reservados do MQTT

   Você pode usar o cliente de teste MQTT para inserir os nomes dos tópicos reservados MQTT do Device Shadow e inscrevê-los para receber atualizações ao executar o`shadow.py`App de amostra. Para inscrever-se nos tópicos:

   a. NoCliente de teste MQTTnoAWS IoTconsole, escolhaAssinar um tópico.
   b. NoFiltro de tópicosseção, insira:$aws/things/***`thingName`***/shadow/update/ #. Aqui,`thingname`É o nome da coisa de recurso do que você criou anteriormente (por exemplo,`My_light_bulb`).
   c. Mantenha os valores padrão para as configurações adicionais e, em seguida, escolhaSubscribe.

   Ao usar o#curinga na assinatura do tópico, você pode assinar vários tópicos do MQTT ao mesmo tempo e observar todas as mensagens que são trocadas entre o dispositivo e seu Shadow em uma única janela. Para obter mais informações sobre os caracteres curinga e o uso deles, consulteTópicos MQTT (p. 99).
3. Execução do**`shadow.py`**programa de amostra e observar mensagens

   Na janela da linha de comando do Raspberry Pi, se você desconectou o programa, execute o aplicativo de amostra novamente e assista às mensagens noCliente de teste MQTTnoAWS IoTconsole.

   a. Execute o comando a seguir para reiniciar o programa de exemplo. Substituir***`your-iot-thing-Name`***e***`your-iot-endpoint`***com os nomes doAWS IoTcoisa que você criou anteriormente (por exemplo,`My_light_bulb`) e o endpoint para interagir com o dispositivo.

      ```
      cd ~/aws-iot-device-sdk-python-v2/samples
      ```

```
python3 shadow.py --ca_file ~/certs/Amazon-root-CA-1.pem --cert ~/certs/
device.pem.crt --key ~/certs/private.pem.key --endpoint your-iot-endpoint --
thing_name your-iot-thing-name
```

O`shadow.py`aplicativo de amostra, em seguida, executa e recupera o estado de sombra atual. Se você excluiu a sombra ou limpou os estados atuais, o programa definirá o valor atual como`off`E, em seguida, solicita que você insira um`desired`value.

```
Connecting to a3qEXAMPLEffp-ats.iot.us-west-2.amazonaws.com with client ID
 'test-0c8ae2ff-cc87-49d2-a82a-ae7ba1d0ca5a'...
Connected!
Subscribing to Delta events...
Subscribing to Update responses...
Subscribing to Get responses...
Requesting current shadow state...
Launching thread to read user input...
Finished getting initial shadow state.
Shadow document lacks 'color' property. Setting defaults...
Changed local shadow value to 'off'.
Updating reported shadow value to 'off'...
Update request published.
Finished updating reported shadow value to 'off'...
Enter desired value:
```

Por outro lado, se o programa estiver em execução e você o reiniciou, você verá o valor de cor mais recente relatado no terminal. No cliente de teste MQTT, você verá uma atualização para os tópicos$aws/things/***thingName***/sombrea/gete$aws/things/***thingName***/shadow/get/accepted.

Suponha que a última cor relatada tenha sido`green`. A seguir, você verá o conteúdo do$aws/things/***thingName***/shadow/get/acceptedArquivo JSON.

```
{
"state": {
  "desired": {
    "welcome": "aws-iot",
    "color": "green"
  },
  "reported": {
    "welcome": "aws-iot",
    "color": "green"
  }
},
"metadata": {
  "desired": {
    "welcome": {
      "timestamp": 1620156892
    },
    "color": {
      "timestamp": 1620161643
    }
  },
  "reported": {
    "welcome": {
      "timestamp": 1620156892
    },
    "color": {
      "timestamp": 1620161643
    }
  }
},
"version": 10,
"timestamp": 1620173908
```

```
}
```

b. Digite um`desired`valor no terminal, como`yellow`. O`shadow.py`aplicativo de amostra responde e exibe as seguintes mensagens no terminal que mostram a alteração na`reported`valor para`yellow`.

```
Enter desired value:
yellow
Changed local shadow value to 'yellow'.
Updating reported shadow value to 'yellow'...
Update request published.
Finished updating reported shadow value to 'yellow'.
```

NoCliente de teste MQTTnoAWS IoTconsole, emAssinaturas, você vê que os tópicos a seguir receberam uma mensagem:

- $aws/things/***thingName***/shadow/update: mostra que ambos`desired`e`updated`valores mudam para a cor`yellow`.
- $aws/things/***thingName***/shadow/update/accepted: mostra os valores atuais do`desired`e`reported`estados e seus metadados e informações de versão.
- $aws/things/***thingName***/shadow/update/documents: mostra os valores anterior e atual do`desired`e`reported`estados e seus metadados e informações de versão.

Como o documento$aws/things/***thingName***/shadow/update/documentstambém contém informações contidas nos outros dois tópicos, podemos revisá-las para ver as informações do estado. O estado anterior mostra o valor relatado definido como`green`, suas informações de metadados e versão e o estado atual que mostra o valor relatado atualizado para`yellow`.

```
{
"previous": {
  "state": {
    "desired": {
      "welcome": "aws-iot",
      "color": "green"
    },
    "reported": {
      "welcome": "aws-iot",
      "color": "green"
    }
  },
  "metadata": {
    "desired": {
      "welcome": {
        "timestamp": 1617297888
      },
      "color": {
        "timestamp": 1617297898
      }
    },
    "reported": {
      "welcome": {
        "timestamp": 1617297888
      },
      "color": {
        "timestamp": 1617297898
      }
    }
  },
  "version": 10
```

```
},
"current": {
  "state": {
    "desired": {
      "welcome": "aws-iot",
      "color": "yellow"
    },
    "reported": {
      "welcome": "aws-iot",
      "color": "yellow"
    }
  },
  "metadata": {
    "desired": {
      "welcome": {
        "timestamp": 1617297888
      },
      "color": {
        "timestamp": 1617297904
      }
    },
    "reported": {
      "welcome": {
        "timestamp": 1617297888
      },
      "color": {
        "timestamp": 1617297904
      }
    }
  },
  "version": 11
},
"timestamp": 1617297904
}
```

c. Agora, se você digitar outro`desired`valor, você vê outras alterações no`reported`valores e atualizações de mensagens recebidas por esses tópicos. O número da versão também aumenta em 1. Por exemplo, se você inserir o valor`green`, o estado anterior relata o valor`yellow`e o estado atual relata o valor`green`.

4. Editar documento Shadow para observar eventos delta

   Para observar alterações no tópico delta, edite o documento Shadow naAWS IoTconsole do . Por exemplo, você pode alterar o`desired`valor para a cor`red`. Para fazer isso, noAWS IoTConsole, escolhaEditee depois defina o`desired`valor para vermelho no JSON, mantendo o`reported`Valor definido como`green`. Antes de salvar a alteração, mantenha o terminal aberto, pois você verá a mensagem delta relatada no terminal.

```
{
"desired": {
  "welcome": "aws-iot",
  "color": "red"
},
"reported": {
  "welcome": "aws-iot",
  "color": "green"
}
}
```

   O`shadow.py`aplicativo de amostra responde a essa alteração e exibe mensagens no terminal indicando o delta. No cliente de teste MQTT, o`update`tópicos receberão uma mensagem mostrando as alterações na`desired`e`reported`Valores.

Você também vê que o tópico$aws/things/***thingName***/shadow/update/deltarecebeu uma mensagem. Para ver a mensagem, escolha este tópico, que está listado emAssinaturas.

```
{
"version": 13,
"timestamp": 1617318480,
"state": {
  "color": "red"
},
"metadata": {
  "color": {
    "timestamp": 1617318480
  }
}
}
```

Etapa 3: Solucionar problemas com interações Device Shadow

Quando você executa o aplicativo de amostra Shadow, você pode encontrar problemas com a observação de interações com o serviço Device Shadow.

Se o programa for executado com sucesso e solicitar que você insira um`desired`valor, você deve ser capaz de observar as interações Device Shadow usando o documento Shadow e o cliente de teste MQTT, conforme descrito anteriormente. No entanto, se você não conseguir ver as interações, veja algumas coisas que você pode verificar:

- Verifique o nome da coisa e sua sombra noAWS IoTconsole

  Se você não vir as mensagens no documento Shadow, revise o comando e certifique-se de que ele corresponda ao nome da coisa naAWS IoTconsole. Você também pode verificar se você tem uma sombra clássica escolhendo seu recurso e, em seguida, escolhendoSombras. Este tutorial se concentra principalmente em interações com a sombra clássica.

  Você também pode confirmar se o dispositivo usado está conectado à Internet. NoAWS IoTconsole, escolha a coisa que você criou anteriormente e, em seguida, escolhaInteragir. Na página de detalhes da coisa, você deve ver uma mensagem aqui dizendo:`This thing already appears to be connected.`
- Verifique os tópicos reservados do MQTT nos quais você se inscreveu

  Se você não vir as mensagens aparecerem no cliente de teste MQTT, verifique se os tópicos que você se inscreveu estão formatados corretamente. Tópicos do MQTT Device Shadow têm um formato$aws/things/***thingName***/shadow/e pode ter`update`,`get`, ou`delete`Seguindo-o dependendo das ações que você quer executar na sombra. Este tutorial usa o tópico$aws/things/***thingName***/sombre/ #portanto, certifique-se de inseri-lo corretamente ao se inscrever no tópico noFiltro de tópicosSeção do cliente de teste.

  Ao digitar o nome do tópico, certifique-se de que o*thingName*É o mesmo que o nome doAWS IoTcoisa que você criou anteriormente. Você também pode se inscrever em tópicos adicionais do MQTT para ver se uma atualização foi executada com sucesso. Por exemplo, você pode se inscrever no tópico$aws/things/***thingName***/shadow/update/rejectedpara receber uma mensagem sempre que uma solicitação de atualização falhou para que você possa depurar problemas de conexão. Para obter mais informações sobre os tópicos reservados, consultethe section called “Tópicos de sombra” (p. 114)eTópicos MQTT do Device Shadow (p. 663).

Etapa 4: Analise os resultados e as próximas etapas

Neste tutorial, você aprendeu a:

- Usar `ashadow.py`aplicativo de amostra para especificar estados desejados e atualizar o estado atual da sombra.
- Edite o documento Shadow para observar eventos delta e como `oshadow.py`aplicativo de amostra responde a ele.
- Use o cliente de teste MQTT para assinar tópicos de sombra e observar atualizações ao executar o programa de amostra.

Próximas etapas

Você pode se inscrever em tópicos reservados MQTT adicionais para observar as atualizações do aplicativo shadow. Por exemplo, se você assinar apenas o tópico$aws/things/***`thingName`***/shadow/update/accepted, você verá somente as informações de estado atual quando uma atualização for executada com sucesso.

Você também pode se inscrever em tópicos de sombra adicionais para depurar problemas ou saber mais sobre as interações Device Shadow e também depurar quaisquer problemas com as interações Device Shadow. Para obter mais informações, consulte the section called “Tópicos de sombra” (p. 114) e Tópicos MQTT do Device Shadow (p. 663).

Você também pode optar por estender seu aplicativo usando sombras nomeadas ou usando hardware adicional conectado ao Raspberry Pi para os LEDs e observar alterações em seu estado usando mensagens enviadas do terminal.

Para obter mais informações sobre o serviço Device Shadow e o uso do serviço em dispositivos, aplicativos e serviços, consulteServiço Device Shadow do AWS IoT (p. 633),Usar sombras em dispositivos (p. 637), eUsar sombras em aplicativos e serviços (p. 640).

Tutorial: Criação de um autorizador personalizado paraAWS IoT Core

Este tutorial demonstra as etapas para criar, validar e usar a autenticação personalizada usando oAWS CLI. Opcionalmente, usando este tutorial, você pode usar o Postman para enviar dados paraAWS IoT Coreusando a API de publicação HTTP.

Este tutorial mostra como criar uma função do Lambda de amostra que implementa a lógica de autorização e autenticação e um autorizador personalizado usando ocreate-authorizerchamada com assinatura de token ativada. O autorizador é então validado usando otest-invoke-authorizere, finalmente, você pode enviar dados paraAWS IoT Coreusando a API de publicação HTTP para um tópico de teste do MQTT. A solicitação de amostra especificará o autorizador a ser invocado usando `ox-amz-customauthorizer-name`cabeçalho e passe o nome da chave do token `ex-amz-customauthorizer-signature`Nos cabeçalhos da solicitação.

O que você aprenderá neste tutorial:

- Como criar uma função do Lambda para ser um manipulador de autorizador personalizado
- Como criar um autorizador personalizado usando oAWS CLICom assinatura de token ativada
- Como testar seu autorizador personalizado usando otest-invoke-authorizerordem
- Como publicar um tópico MQTT usandoCarteiroe valide a solicitação com seu autorizador personalizado

A conclusão deste tutorial requer cerca de 60 minutos.

Neste tutorial, você irá:

- Etapa 1: Crie uma função do Lambda para seu autorizador personalizado (p. 245)
- Etapa 2: Crie um key pair pública e privada para o autorizador personalizado (p. 247)
- Etapa 3: Criar um recurso de autorizador do cliente e sua autorização (p. 247)
- Etapa 4: Teste o autorizador chamando test-invoke-authorizer (p. 250)
- Etapa 5: Teste a publicação da mensagem MQTT usando o Postman (p. 251)
- Etapa 6: Exibir mensagens no cliente de teste MQTT (p. 253)
- Etapa 7: Analise os resultados e as próximas etapas (p. 254)
- Etapa 8 Limpar (p. 254)

Antes de começar este tutorial, certifique-se de que você tenha:

- Configurar seuConta da AWS (p. 19)

  Você precisará doConta da AWSeAWS IoTConsole para concluir este tutorial.

  A conta que você usa para este tutorial funciona melhor quando inclui pelo menos essesAWSpolíticas gerenciadas:

  - `IAMFullAccess`
  - `AWSIoTFullAccess`
  - `AWSLambda_FullAccess`

  > **Important**
  >
  > As políticas do IAM usadas neste tutorial são mais permissivas do que você deveria seguir em uma implementação de produção. Em um ambiente de produção, certifique-se de que suas políticas de conta e recursos concedam somente as permissões necessárias.
  > Quando você criar políticas do IAM para produção, determine o acesso que os usuários e funções precisam e crie as políticas que lhes permitam executar apenas essas tarefas.
  > Para obter mais informações, consultePráticas recomendadas de segurança no IAM

- Instalado oAWS CLI

  Para obter informações sobre como instalar oAWS CLI, consulteInstalar oAWSCLI. Este tutorial requerAWS CLIversão`aws-cli/2.1.3 Python/3.7.4 Darwin/18.7.0 exe/x86_64`ou posterior.

- Ferramentas OpenSSL

  Use os exemplos neste tutorialLibreSSL 2.6.5. Você também pode usarOpenSSL v1.1.1iferramentas para este tutorial.

- Analise oAWS Lambdavisão geral

  Se você ainda não usou oAWS Lambdaantes, revejaAWS LambdaeConceitos básicos do LambdaPara aprender seus termos e conceitos.

- Analisado como criar solicitações no Postman

  Para obter mais informações, consulteSolicitações de construção.

- Autorizadores personalizados removidos do tutor anterior

  SuasConta da AWSsó pode ter um número limitado de autorizadores personalizados configurados de uma só vez. Para obter informações sobre como remover um autorizador personalizado, consultethe section called “Etapa 8 Limpar” (p. 254).

Etapa 1: Crie uma função do Lambda para seu autorizador personalizado

Autenticação personalizada emAWS IoT Coreusarecursos do autorizadorque você cria para autenticar e autorizar clientes. A função que você criará nesta seção autentica e autoriza os clientes à medida que eles se conectam aoAWS IoT Coree acessarAWS IoTrecursos da AWS.

A função do Lambda faz o seguinte:

- Se uma solicitação vier detest-invoke-authorizer, ele retorna uma política do IAM com um`Deny`Ação .
- Se uma solicitação vier do Passport usando HTTP e o`actionToken`parâmetro tem um valor de`allow`, ele retorna uma política do IAM com um`Allow`Ação . Caso contrário, ele retornará uma política do IAM com um`Deny`Ação .

Para criar a função do Lambda para seu autorizador personalizado

1. NoLambdaConsole do, abertoFunções.
2. Escolha Create function (Criar função).
3. ConfirmarAuthor from scratché selecionado.
4. Em Basic information:
   a. Em Function name (Nome da função), insira **`custom-auth-function`**.
   b. DentroTempo de execução, confirmeNode.js 14.x
5. Escolha Create function (Criar função).

   O Lambda cria uma função Node.js e uma função de execução que concede à função permissão para fazer upload de logs. A função do Lambda assume a função de execução quando você invoca sua função e usa a função de execução para criar credenciais para oAWSSDK e para ler dados de fontes de eventos.
6. Para ver o código e a configuração da função naAWS Cloud9editor, escolhafunção custom-auth-Na janela do designer e depois escolhaindex.jsNo painel de navegação do editor.

   Para linguagens de script como Node.js, o Lambda inclui uma função básica que retorna uma resposta de sucesso. Você pode usar oAWS Cloud9Para editar sua função, desde que o código-fonte não exceda 3 MB.
7. Substitua oindex.jsCode no editor com o código a seguir:

```
// A simple Lambda function for an authorizer. It demonstrates
// How to parse a CLI and Http password to generate a response.

exports.handler = function(event, context, callback) {

    //Http parameter to initiate allow/deny request
    const HTTP_PARAM_NAME='actionToken';
    const ALLOW_ACTION = 'Allow';
    const DENY_ACTION = 'Deny';

    //Event data passed to Lambda function
    var event_str = JSON.stringify(event);
    console.log('Complete event :'+ event_str);

    //Read protocolData from the event json passed to Lambda function
    var protocolData = event.protocolData;
    console.log('protocolData value---> ' + protocolData);

    //Get the dynamic account ID from function's ARN to be used
```

```
    // as full resource for IAM policy
    var ACCOUNT_ID = context.invokedFunctionArn.split(":")[4];
    console.log("ACCOUNT_ID---"+ACCOUNT_ID);

    //Get the dynamic region from function's ARN to be used
    // as full resource for IAM policy
    var REGION = context.invokedFunctionArn.split(":")[3];
    console.log("REGION---"+REGION);

    //protocolData data will be undefined if testing is done via CLI.
    // This will help to test the set up.
    if (protocolData === undefined) {

        //If CLI testing, pass deny action as this is for testing purpose only.
        console.log('Using the test-invoke-authorizer cli for testing only');
        callback(null, generateAuthResponse(DENY_ACTION,ACCOUNT_ID,REGION));

    } else{

        //Http Testing from Postman
        //Get the query string from the request
        var queryString = event.protocolData.http.queryString;
        console.log('queryString values -- ' + queryString);
        /*         global URLSearchParams       */
        const params = new URLSearchParams(queryString);
        var action = params.get(HTTP_PARAM_NAME);

        if(action!=null && action.toLowerCase() === 'allow'){

            callback(null, generateAuthResponse(ALLOW_ACTION,ACCOUNT_ID,REGION));

        }else{

            callback(null, generateAuthResponse(DENY_ACTION,ACCOUNT_ID,REGION));

        }

    }

};

// Helper function to generate the authorization IAM response.
var generateAuthResponse = function(effect,ACCOUNT_ID,REGION) {

    var full_resource = "arn:aws:iot:"+ REGION + ":" + ACCOUNT_ID + ":*";
    console.log("full_resource---"+full_resource);

    var authResponse = {};
    authResponse.isAuthenticated = true;
    authResponse.principalId = 'principalId';

    var policyDocument = {};
    policyDocument.Version = '2012-10-17';
    policyDocument.Statement = [];
    var statement = {};
    statement.Action = 'iot:*';
    statement.Effect = effect;
    statement.Resource = full_resource;
    policyDocument.Statement[0] = statement;
    authResponse.policyDocuments = [policyDocument];
    authResponse.disconnectAfterInSeconds = 3600;
    authResponse.refreshAfterInSeconds = 600;

    console.log('custom auth policy function called from http');
    console.log('authResponse --> ' + JSON.stringify(authResponse));
    console.log(authResponse.policyDocuments[0]);
```

```
    return authResponse;
}
```

8. Escolha Implantar.
9. DepoisAlterações implantadasaparece acima do editor:
   a. Rolar até oVisão geral da funçãoseção acima do editor.
   b. Copie oNome de recurso da Amazon da funçãoE salve-o para usá-lo posteriormente neste tutorial.
10. Teste a função do .
   a. Selecione oTesteGuia.
   b. Usando as configurações de teste padrão, escolhaInvoke.
   c. Se o teste for bem-sucedido, noResultados da execução, abra oDetalhesexibição. Você deve ver o documento de política que a função retornou.

      Se o teste falhar ou você não vir um documento de política, revise o código para localizar e corrigir os erros.

Etapa 2: Crie um key pair pública e privada para o autorizador personalizado

Seu autorizador personalizado requer uma chave pública e privada para autenticá-la. Os comandos desta seção usam ferramentas OpenSSL para criar esse key pair.

Para criar o key pair pública e privada para o autorizador personalizado

1. Crie o arquivo de chave privada.

```
openssl genrsa -out private-key.pem 4096
```

2. Verifique o arquivo de chave privada que você acabou de criar.

```
openssl rsa -check -in private-key.pem -noout
```

   Se o comando não exibir nenhum erro, o arquivo de chave privada será válido.
3. Crie o arquivo de chave pública.

```
openssl rsa -in private-key.pem -pubout -out public-key.pem
```

4. Verifique o arquivo de chave pública.

```
openssl pkey -inform PEM -pubin -in public-key.pem -noout
```

   Se o comando não exibir nenhum erro, o arquivo de chave pública será válido.

Etapa 3: Criar um recurso de autorizador do cliente e sua autorização

OAWS IoTautorizador personalizado é o recurso que une todos os elementos criados nas etapas anteriores. Nesta seção, você criará um recurso autorizador personalizado e lhe dará permissão para

executar a função do Lambda criada anteriormente. Você pode criar um recurso autorizador personalizado usando oAWS IoTConsole do, oAWS CLI, ou oAWSAPI.

Para este tutorial, você só precisará criar um autorizador personalizado. Esta seção descreve como criar usando oAWS IoTconsole e oAWS CLI, para que você possa usar o método mais conveniente para você. Não há diferença entre os recursos do autorizador personalizado criados por nenhum dos métodos.

Criar um recurso de autorizador do cliente

Escolha uma destas opções para criar o recurso do autorizador personalizado

- Crie um autorizador personalizado usando oAWS IoTconsole (p. 248)
- Crie um autorizador personalizado usando oAWS CLI (p. 249)

Para criar um autorizador personalizado (console)

1. Abrir oPágina autorizador personalizado doAWS IoTconsolee escolhaCriar.
2. DentroCriar autorizador personalizado:
   a. DentroNomeie seu autorizador personalizado, insira**`my-new-authorizer`**.
   b. DentroFunção autorizador, escolha a função do Lambda criada anteriormente.
   c. DentroValidação de token - opcional:
      i. CheckHabilitar assinatura de token.
      ii. DentroNome do cabeçalho do token (opcional), insira**`tokenKeyName`**.
      iii. DentroNome da chave, insira:**`FirstKey`**.
      iv. DentroValor, insira o conteúdo do`public-key.pem`file. Certifique-se de incluir as linhas do arquivo com`-----BEGIN PUBLIC KEY-----`e`-----END PUBLIC KEY-----`e não adicione nem remova nenhum arquivo de linha, retornos de carro ou outros caracteres do conteúdo do arquivo. A string que você insere deve ser semelhante a este exemplo.

```
-----BEGIN PUBLIC KEY-----
MIICIjANBgkqhkiG9w0BAQEFAAOCAg8AMIICCgKCAgEAvEBzOk4vhN+3LgslvEWt
sLCqNmt5Damas3bmiTRvq2gjRJ6KXGTGQChqArAJwL1a9dkS9+maaXC3vc6xzx9z
QPu/vQOe5tyzz1MsKdmtFGxMqQ3qjEXAMPLEOmqyUKPP5mff58k6ePSfXAnzBH0q
lg2HioefrpU5OSAnpuRAjYKofKjbc2Vrn6N2G7hV+IfTBvCElf0csalS/Rk4phD5
oa4Y0GHISRnevypg5C8n9Rrz91PWGqP6M/q5DNJJXjMyleG92hQgu1N696bn5Dw8
FhedszFa6b2x6xrItZFzewNQkPMLMFhNrQIIyvshtT/F1LVCS5+v8AQ8UGGDfZmv
QeqAMAF7WgagDMXcfgKSVU8yid2sIm56qsCLMvD2Sq8Lgzpey9N5ON1o1Cvldwvc
KrJJtgwW6hVqRGuShnownLpgG86M6neZ5sRMbVNZO8OzcobLngJ0Ibw9KkcUdklW
gvZ6HEJqBY2XE70iEXAMPLETPHzhqvK6Ei1HGxpHsXx6BNft582J1VpgYjXha8oa
/NN7l7Zbj/euAb41IVtmX8JrD9z613d1iM5L8HluJlUzn62Q+VeNV2tdA7MfPfMC
8btGYladFAnitThaz6+F0VSBJPu7pZQoLnqyEp5zLMtF+kFl2yOBmGAP0RBivRd9
JWBUCG0bqcLQPeQyjbXSOfUCAwEAAQ==
-----END PUBLIC KEY-----
```

3. CheckAtivar o autorizador.
4. SelecioneCriar autorizador.
5. Se o recurso autorizador personalizado tiver sido criado, você verá a lista de autorizadores personalizados e seu novo autorizador personalizado deverá aparecer na lista e você poderá continuar para a próxima seção para testá-lo.

   Se você vir um erro, revise o erro e tente criar seu autorizador personalizado novamente e verifique novamente as entradas. Observe que cada recurso do autorizador personalizado deve ter um nome exclusivo.

Para criar um autorizador personalizado (AWS CLI)

1. Substitua seus valores por`authorizer-function-arn`e`token-signing-public-keys`E execute o seguinte comando:

```
aws iot create-authorizer \
--authorizer-name "my-new-authorizer" \
--token-key-name "tokenKeyName" \
--status ACTIVE \
--no-signing-disabled \
--authorizer-function-arn "arn:aws:lambda:Region:57EXAMPLE833:function:custom-auth-
function" \
--token-signing-public-keys FirstKey="-----BEGIN PUBLIC KEY-----
MIICIjANBgkqhkiG9w0BAQEFAAOCAg8AMIICCgKCAgEAvEBzOk4vhN+3LgslvEWt
sLCqNmt5Damas3bmiTRvq2gjRJ6KXGTGQChqArAJwL1a9dkS9+maaXC3vc6xzx9z
QPu/vQOe5tyzz1MsKdmtFGxMqQ3qjEXAMPLEOmqyUKPP5mff58k6ePSfXAnzBH0q
lg2HioefrpU5OSAnpuRAjYKofKjbc2Vrn6N2G7hV+IfTBvCElf0csalS/Rk4phD5
oa4Y0GHISRnevypg5C8n9Rrz91PWGqP6M/q5DNJJXjMyleG92hQgu1N696bn5Dw8
FhedszFa6b2x6xrItZFzewNQkPMLMFhNrQIIyvshtT/F1LVCS5+v8AQ8UGGDfZmv
QeqAMAF7WgagDMXcfgKSVU8yid2sIm56qsCLMvD2Sq8Lgzpey9N5ON1o1Cvldwvc
KrJJtgwW6hVqRGuShnownLpgG86M6neZ5sRMbVNZO8OzcobLngJ0Ibw9KkcUdklW
gvZ6HEJqBY2XE70iEXAMPLETPHzhqvK6Ei1HGxpHsXx6BNft582J1VpgYjXha8oa
/NN7l7Zbj/euAb41IVtmX8JrD9z613d1iM5L8HluJlUzn62Q+VeNV2tdA7MfPfMC
8btGYladFAnitThaz6+F0VSBJPu7pZQoLnqyEp5zLMtF+kFl2yOBmGAP0RBivRd9
JWBUCG0bqcLQPeQyjbXSOfUCAwEAAQ==
-----END PUBLIC KEY-----"
```

Onde:

- O`authorizer-function-arn`O valor é o nome de recurso da Amazon (ARN) da função do Lambda que você criou para o autorizador personalizado.
- O`token-signing-public-keys`O valor inclui o nome da chave,**`FirstKey`**e o conteúdo do`public-key.pem`file. Certifique-se de incluir as linhas do arquivo com`-----BEGIN PUBLIC KEY-----`e`-----END PUBLIC KEY-----`e não adicione nem remova nenhum arquivo de linha, retornos de carro ou outros caracteres do conteúdo do arquivo.

  Nota: tenha cuidado ao inserir a chave pública, pois qualquer alteração no valor da chave pública a torna inutilizável.

2. Se o autorizador personalizado for criado, o comando retornará o nome e o ARN do novo recurso, como o seguinte.

```
{
    "authorizerName": "my-new-authorizer",
    "authorizerArn": "arn:aws:iot:Region:57EXAMPLE833:authorizer/my-new-authorizer"
}
```

Save the`authorizerArn`Valor para uso na próxima etapa.

Lembre-se de que cada recurso do autorizador personalizado deve ter um nome exclusivo.

Autorizar o recurso autorizador personalizado

Nesta seção, você concederá permissão ao recurso autorizador personalizado que acabou de criar permissão para executar a função do Lambda

Conceder permissão à sua função do Lambda usando oAWS CLI

1. Depois de inserir seus valores, digite o comando a seguir. Observe que `ostatement-id`O valor deve ser exclusivo. Substituir*`Id-1234`*com outro valor se você tiver executado este tutorial antes ou se você receber um`ResourceConflictException`(Nenhum console do Windows encontrado. Você está executando o cmd.exe?".

```
aws lambda add-permission  \
--function-name "custom-auth-function" \
--principal "iot.amazonaws.com" \
--action "lambda:InvokeFunction" \
--statement-id "Id-1234" \
--source-arn authorizerArn
```

2. Se o comando for bem-sucedido, ele retornará uma instrução de permissão, como este exemplo. Você pode continuar na próxima seção para testar o autorizador personalizado.

```
{
    "Statement": "{\"Sid\":\"Id-1234\",\"Effect\":\"Allow\",\"Principal\":{\"Service
\":\"iot.amazonaws.com\"},\"Action\":\"lambda:InvokeFunction\",\"Resource\":
\"arn:aws:lambda:Region:57EXAMPLE833:function:custom-auth-function\",\"Condition\":
{\"ArnLike\":{\"AWS:SourceArn\":\"arn:aws:lambda:Region:57EXAMPLE833:function:custom-
auth-function\"}}}"
}
```

Se o comando não for bem-sucedido, ele retornará um erro, como este exemplo. Você precisará revisar e corrigir o erro antes de continuar.

```
An error occurred (AccessDeniedException) when calling the AddPermission operation:
 User: arn:aws:iam::57EXAMPLE833:user/EXAMPLE-1 is not authorized to perform:
 lambda:AddPer
mission on resource: arn:aws:lambda:Region:57EXAMPLE833:function:custom-auth-function
```

Etapa 4: Teste o autorizador chamando test-invoke-authorizer

Com todos os recursos definidos, nesta seção, você chamará test-invoke-authorizer a partir da linha de comando para testar a aprovação de autorização.

Observe que, ao invocar o autorizador da linha de comando,`protocolData`não está definido, portanto, o autorizador sempre retornará um documento DENY. Este teste, no entanto, confirma que seu autorizador personalizado e a função do Lambda estão configurados corretamente, mesmo que não teste totalmente a função do Lambda.

Para testar seu autorizador personalizado e sua função do Lambda usando oAWS CLI

1. No diretório que contém `oprivate-key.pem`Você criou em uma etapa anterior, execute o seguinte comando:

```
echo -n "tokenKeyValue" | openssl dgst -sha256 -sign private-key.pem | openssl base64 -
A
```

Esse comando cria uma string de assinatura para uso na próxima etapa. A string de assinatura é semelhante ao seguinte:

```
dBwykzlb+fo+JmSGdwoGr8dyC2qB/IyLefJJr+rbCvmu9Jl4KHAA9DG+V
+MMWu09YSA86+64Y3Gt4tOykpZqn9mn
```

```
VB1wyxp+0bDZh8hmqUAUH3fwi3fPjBvCa4cwNuLQNqBZzbCvsluv7i2IMjEg
+CPY0zrWt1jr9BikgGPDxWkjaeeh
bQHHTo357TegKs9pP30Uf4TrxypNmFswA5k7QIc01n4bIyRTm90OyZ94R4bdJsHNig1JePgnuOBvMGCEFE09jGjj
szEHfgAUAQIWXiVGQj16BU1xKpTGSiTAwheLKUjITOEXAMPLECK3aHKYKY
+d1vTvdthKtYHBq8MjhzJ0kggbt29V
QJCb8RilN/P5+vcVniSXWPplyB5jkYs9UvG08REoy64AtizfUhvSul/r/F3VV8ITtQp3aXiUtcspACi6ca
+tsDuX
f3LzCwQQF/YSUy02u5XkWn+sto6KCkpNlkD0wU8gl3+kOzxrthnQ8gEajd5Iylx230iqcXo3osjPha7JDyWM5o
+K
EWckTe91I1mokDr5sJ4JXixvnJTVSx1li49IalW4en1DAkc1a0s2U2UNm236EXAMPLELotyh7h
+flFeloZlAWQFH
xRlXsPqiVKS1ZIUClaZWprh/orDJplpiWfBgBIOgokJIDGP9gwhXIIk7zWrGmWpMK9o=
```

Copie essa string de assinatura para usar na próxima etapa. Tenha cuidado para não incluir caracteres extras ou deixar de fora.

2. Neste comando, substitua`token-signature`valor com a string de assinatura da etapa anterior e execute este comando para testar seu autorizador.

```
aws iot test-invoke-authorizer \
--authorizer-name my-new-authorizer \
--token tokenKeyValue \
--token-signature dBwykzlb+fo+JmSGdwoGr8dyC2qB/IyLefJJr
+rbCvmu9Jl4KHAA9DG+V+MMWu09YSA86+64Y3Gt4tOykpZqn9mnVB1wyxp
+0bDZh8hmqUAUH3fwi3fPjBvCa4cwNuLQNqBZzbCvsluv7i2IMjEg
+CPY0zrWt1jr9BikgGPDxWkjaeehbQHHTo357TegKs9pP30Uf4TrxypNmFswA5k7QIc01n4bIyRTm90OyZ94R4bdJsHNig1JePg
+d1vTvdthKtYHBq8MjhzJ0kggbt29VQJCb8RilN/P5+vcVniSXWPplyB5jkYs9UvG08REoy64AtizfUhvSul/
r/F3VV8ITtQp3aXiUtcspACi6ca+tsDuXf3LzCwQQF/YSUy02u5XkWn
+sto6KCkpNlkD0wU8gl3+kOzxrthnQ8gEajd5Iylx230iqcXo3osjPha7JDyWM5o
+KEWckTe91I1mokDr5sJ4JXixvnJTVSx1li49IalW4en1DAkc1a0s2U2UNm236EXAMPLELotyh7h
+flFeloZlAWQFHxRlXsPqiVKS1ZIUClaZWprh/orDJplpiWfBgBIOgokJIDGP9gwhXIIk7zWrGmWpMK9o=
```

Se o comando for bem-sucedido, ele retornará as informações geradas pela função de autorizador do cliente, como este exemplo.

```
{
    "isAuthenticated": true,
    "principalId": "principalId",
    "policyDocuments": [
        "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Action\":\"iot:*\",\"Effect\":
\"Deny\",\"Resource\":\"arn:aws:iot:Region:57EXAMPLE833:*\"}]}"
    ],
    "refreshAfterInSeconds": 600,
    "disconnectAfterInSeconds": 3600
}
```

Se o comando retornar um erro, revise o erro e verifique novamente os comandos usados nesta seção.

Etapa 5: Teste a publicação da mensagem MQTT usando o Postman

1. Para obter o endpoint de dados do dispositivo a partir da linha de comando, chamedescribe-endpointcomo mostrado aqui

```
aws iot describe-endpoint --output text --endpoint-type iot:Data-ATS
```

Salve este endereço para uso como o*device_data_endpoint_address*Em uma etapa posterior.

2. Abra uma nova janela Postman e crie uma nova solicitação HTTP POST.
   a. No seu computador, abra o aplicativo Postman.
   b. No carteiro, noArquivo, selecioneNovo....
   c. NoNovo, selecioneSolicitação.
   d. Em Save request,
      i. DentroNome da solicitaçãoentrar**`Custom authorizer test request`**.
      ii. DentroSelecione uma coleção ou pasta para salvar em:escolha ou crie uma coleção na qual salvar essa solicitação.
      iii. SelecioneSave to***`collection_name`***.
3. Crie a solicitação POST para testar seu autorizador personalizado.
   a. No seletor de método de solicitação ao lado do campo URL, escolhaPUBLICAR.
   b. No campo URL, crie o URL para sua solicitação usando o seguinte URL com o*`device_data_endpoint_address`*dodescribe-endpointcomando em uma etapa anterior.

```
https://device_data_endpoint_address:443/topics/test/cust-auth/topic?
qos=0&actionToken=allow
```

      Observe que esse URL inclui o`actionToken=allow`parâmetro de consulta que informará à função do Lambda para retornar um documento de política que permita acesso aAWS IoT. Depois de inserir o URL, os parâmetros de consulta também aparecem noParamsGuia do Postman.
   c. NoAutenticaçãoguia, na guiaTipocampo, escolhaNenhuma autenticação.
   d. Na guia Headers:
      i. Se houver umHostchave que está marcada, desmarque esta.
      ii. Na parte inferior da lista de cabeçalhos, adicione esses novos cabeçalhos e confirme que eles estão verificados. Substitua o**`Host`**valor com seu*`device_data_endpoint_address`*O e a**`x-amz-customauthorizer-signature`**valor com a string de assinatura que você usou com otest-invoke-authorizeCommand na seção anterior.

Chave	Valor
`x-amz-customauthorizer-name`	**`my-new-authorizer`**
`Host`	*`device_data_endpoint_address`*
`tokenKeyName`	**`tokenKeyValue`**
`x-amz-customauthorizer-signature`	*`dbwykzlb+fo+Jmsgdwogr8dyc2qb/ iylefjjr+rbcvmu9jl4khaa9dG+V +mmWu09ysa86+64Y3GT4Toykpzqn9mnvb1wyxp +0BDzh8hmquauh3fwi3fpjbvca4cwnulqnulqNQBZZBC +CPY0ZRWT1JR9BIKGGPDXWKJAEehBQHTO357TEGks9PF xkptgsitawhelkujitoexampleck3ahkyky +d1vtvdthktyhbq8mjhzj0kggbt29vqjcb8riln/ p5+vcvnisxwpplyb5jkys9uvg08reoy64atizfuhvsul R/F3vv8ittQP3axiutcspacic 6CA +TSDUXF3LZCWQQF/YSUY02U5xKWN +STO6KCKPNLKD0WU8GL3+KOZXRTHNQ8GEAJD5iYLX230 +KEWCKTE91i1MOKDR5SJ4JXIXVNJTVSX1LI49IALW4EN H +FLFELOZLAWQFHXRLXSPQiVKS1ZiuclazWPRH/`*

Chave	Valor
	OrdJPLPIWFBGBioGokjidGP9GWHXIIK7ZWRGMWPMK9O=

e. Na guia Corpo:

   i. Na caixa de opção de formato de dados, escolhaRaw.

   ii. Na lista de tipos de dados, escolhaJavaScript.

   iii. No campo de texto, insira esta carga de mensagem JSON para sua mensagem de teste:

```
{
    "data_mode": "test",
    "vibration": 200,
    "temperature": 40
}
```

4. SelecioneEnviarPara enviar a solicitação.

   Se a solicitação tiver sido bem-sucedida, ela retornará:

```
{
    "message": "OK",
    "traceId": "ff35c33f-409a-ea90-b06f-fbEXAMPLE25c"
}
```

   A resposta bem-sucedida indica que seu autorizador personalizado permitiu a conexão comAWS IoTe que a mensagem de teste foi entregue ao corretor emAWS IoT Core.

   Se ele retornar um erro, revise a mensagem de erro, o*device_data_endpoint_address*, a string de assinatura e os outros valores de cabeçalho.

Mantenha essa solicitação no Postman para uso na próxima seção.

Etapa 6: Exibir mensagens no cliente de teste MQTT

Na etapa anterior, você enviou mensagens de dispositivo simuladas paraAWS IoTUsando o Postman. A resposta bem-sucedida indicou que seu autorizador personalizado permitiu a conexão comAWS IoTe que a mensagem de teste foi entregue ao corretor emAWS IoT Core. Nesta seção, você usará o cliente de teste MQTT naAWS IoTconsole para ver o conteúdo da mensagem dessa mensagem como outros dispositivos e serviços podem.

Para ver as mensagens de teste autorizadas pelo autorizador personalizado

1. NoAWS IoTConsole do, abra oCliente de teste MQTT da.
2. NoAssinar o tópico, emFiltro de tópicos, insira**`test/cust-auth/topic`**, que é o tópico da mensagem usado no exemplo do Postman da seção anterior.
3. Escolha Subscribe.

   Mantenha essa janela visível para a próxima etapa.
4. No Postman, na solicitação que você criou para a seção anterior, escolhaEnviar.

   Revise a resposta para garantir que ela tenha sido bem-sucedida. Caso contrário, solucione o erro como a seção anterior descreve.
5. NoCliente de teste MQTT da, você verá uma nova entrada que mostra o tópico da mensagem e, se expandida, a carga útil da mensagem da solicitação que você enviou do Postman.

Se você não vir suas mensagens naCliente de teste MQTT da, aqui estão algumas coisas para verificar:

- Certifique-se de que sua solicitação do Postman seja retornada com sucesso. SeAWS IoTrejeita a conexão e retorna um erro, a mensagem na solicitação não é passada para o corretor de mensagens.
- Certifique-se de que oConta da AWSeRegião da AWSusado para abrir oAWS IoTo console é o mesmo que você está usando na URL do Postman.
- Certifique-se de que você inseriu o tópico corretamente noCliente de teste MQTT da. Observe que o filtro de tópicos diferencia maiúsculas Em caso de dúvida, você também pode se inscrever no#tópico, que se inscreve em todas as mensagens do MQTT que passam pelo corretor de mensagensConta da AWSeRegião da AWSusado para abrir oAWS IoTconsole do .

Etapa 7: Analise os resultados e as próximas etapas

Neste tutorial:

- Você criou uma função do Lambda para ser um manipulador de autorizador personalizado
- Você criou um autorizador personalizado com assinatura de token ativada
- Você testou seu autorizador personalizado usando otest-invoke-authorizerordem
- Você publicou um tópico MQTT usandoCarteiroe valide a solicitação com seu autorizador personalizado
- Você usou oCliente de teste MQTT dapara ver as mensagens enviadas do teste do carteiro

Próximas etapas

Depois de enviar algumas mensagens do Postman para verificar se o autorizador personalizado está funcionando, tente experimentar para ver como a mudança de diferentes aspectos deste tutorial afeta os resultados. Veja aqui alguns exemplos para começar a usar.

- Altere a string de assinatura para que ela não seja mais válida ver como as tentativas de conexão não autorizadas são tratadas. Você deve receber uma resposta de erro, como esta, e a mensagem não deve aparecer naCliente de teste MQTT da.

```
{
    "message": "Forbidden",
    "traceId": "15969756-a4a4-917c-b47a-5433e25b1356"
}
```

- Para saber mais sobre como encontrar erros que podem ocorrer enquanto você está desenvolvendo e usandoAWS IoTregras, consulteMonitorar o AWS IoT (p. 432).

Etapa 8 Limpar

Se você quiser repetir este tutorial, talvez seja necessário remover alguns de seus autorizadores personalizados. SuasConta da AWSpode ter apenas um número limitado de autorizadores personalizados configurados de uma só vez e você pode obter um`LimitExceededException`quando você tenta adicionar um novo sem remover um autorizador personalizado existente.

Para remover um autorizador personalizado (console)

1. Abrir oPágina autorizador personalizado doAWS IoTconsolee na lista de autorizadores personalizados, encontre o autorizador personalizado a ser removido.
2. Abra a página de detalhes do autorizador personalizado e, noAções, selecioneEdite.

3. Desmarque oAtivar o autorizadore, depois, escolhaAtualização.

   Não é possível excluir um autorizador personalizado enquanto ele estiver ativo.
4. Na página de detalhes do autorizador personalizado, abra a guiaAçõesmenu e escolhaExcluir.

Para remover um autorizador personalizado (AWS CLI)

1. Liste os autorizadores personalizados que você instalou e encontre o nome do autorizador personalizado que deseja excluir.

```
aws iot list-authorizers
```

2. Defina o autorizador personalizado como`inactive`executando este comando após a substituição*`Custom_Auth_Name`*com o`authorizerName`do autorizador personalizado a ser excluído.

```
aws iot update-authorizer --status INACTIVE --authorizer-name Custom_Auth_Name
```

3. Exclua o autorizador personalizado executando este comando após a substituição*`Custom_Auth_Name`*com o`authorizerName`do autorizador personalizado a ser excluído.

```
aws iot delete-authorizer --authorizer-name Custom_Auth_Name
```

Tutorial: Monitoramento da umidade do soloAWS IoTe Raspberry Pi

Este tutorial mostra como usar um Raspberry Pi, um sensor de umidade e o AWS IoT para monitorar o nível de umidade do solo de uma planta doméstica ou de um jardim. O Raspberry Pi executa um código que lê o nível de umidade e a temperatura do sensor e envia os dados para o AWS IoT. Você cria uma regra noAWS IoTO que envia um e-mail para um endereço inscrito em um tópico do Amazon SNS quando o nível de umidade fica abaixo de um limite.

> **Note**
>
> Esse tutorial pode não estar atualizado. Algumas referências podem ter sido substituídas desde que esse tópico foi publicado originalmente.

Sumário

- Pré-requisitos (p. 255)
- Configurar o AWS IoT (p. 256)
  - Etapa 1: Crie a política do AWS IoT (p. 256)
  - Etapa 2: Criar a coisa, o certificado e a chave privada do AWS IoT (p. 258)
  - Etapa 3: Criar um tópico do Amazon SNS e uma assinatura (p. 258)
  - Etapa 4: Criar uma regra do AWS IoT para enviar um e-mail (p. 259)
- Configuração do Raspberry Pi e do sensor de umidade (p. 260)

Pré-requisitos

Para concluir este tutorial, você precisa de:

- Uma Conta da AWS.
- Um usuário do IAM com permissões de administrador.
- Um computador de desenvolvimento que executa Windows, macOS, Linux ou Unix para acessar o console do AWS IoT.
- UMARaspberry Pi 3B ou 4Bexecutando o mais recenteSO Raspbian. Para obter instruções de instalação, consulte Installing operating system images no site do Rasberry Pi.
- Um monitor, um teclado, um mouse e uma rede Wi-Fi ou conexão Ethernet para o Raspberry Pi.
- Um sensor de umidade compatível com o Raspberry Pi. O sensor usado neste tutorial é um Adafruit STEMMA I2C Capacitive Moisture Sensor com um JST de 4 pinos com conector de cabo de soquete fêmea.

Configurar o AWS IoT

Para concluir este tutorial, é necessário criar os recursos a seguir. Para conectar um dispositivo ao AWS IoT, crie um item de IoT, um certificado de dispositivo e uma política do AWS IoT.

- Uma coisa do AWS IoT

  Uma coisa representa um dispositivo físico (nesse caso, o Rasberry Pi) e contém metadados estáticos sobre o dispositivo.
- Um certificado de dispositivo.

  Todos os dispositivos devem ter um certificado de dispositivo para se conectar e autenticar com o AWS IoT.
- Uma política do AWS IoT.

  Cada certificado de dispositivo tem uma ou mais políticas do AWS IoT associadas a ele. Essas políticas determinam quais recursos do AWS IoT o dispositivo pode acessar.
- Um certificado CA raiz do AWS IoT.

  Os dispositivos e outros clientes usam um certificado CA raiz do AWS IoT para autenticar o servidor do AWS IoT com o qual estão se comunicando. Para obter mais informações, consulte Autenticação do servidor (p. 295).
- Uma regra do AWS IoT

  Uma regra contém uma consulta e uma ou mais ações de regra. A consulta extrai dados de mensagens do dispositivo para determinar se os dados da mensagem devem ser processados. A ação da regra especifica o que fazer se os dados corresponderem à consulta.
- Um tópico do Amazon SNS e uma assinatura de tópico do.

  A regra escuta dados de umidade do Raspberry Pi. Se o valor estiver abaixo de um limite, ele enviará uma mensagem para o tópico do Amazon SNS. O Amazon SNS envia essa mensagem para todos os endereços de e-mail que assinaram o tópico.

Etapa 1: Crie a política do AWS IoT

Crie uma política do AWS IoT que permita que o Raspberry Pi se conecte e envie mensagens ao AWS IoT.

1. No console do AWS IoT, se um botão Get started (Começar) for exibido, selecione-o. Caso contrário, no painel de navegação, expanda Secure (Proteger) e selecione Policies (Políticas).
2. Se uma caixa de diálogo You don't have any policies yet (Você ainda não tem políticas), selecione Create a policy (Criar uma política). Caso contrário, escolha Create (Criar).

3. Insira um nome para a política do AWS IoT (por exemplo, **`MoistureSensorPolicy`**).
4. Na seção Add statements (Adicionar instruções), substitua a política existente pelo JSON a seguir. Substituir*região*e*conta*Com o seuRegião da AWSeConta da AWSnúmero.

```
{
   "Version": "2012-10-17",
   "Statement": [{
         "Effect": "Allow",
         "Action": "iot:Connect",
         "Resource": "arn:aws:iot:region:account:client/RaspberryPi"
      },
      {
         "Effect": "Allow",
         "Action": "iot:Publish",
         "Resource": [
            "arn:aws:iot:region:account:topic/$aws/things/RaspberryPi/shadow/update",
            "arn:aws:iot:region:account:topic/$aws/things/RaspberryPi/shadow/delete",
            "arn:aws:iot:region:account:topic/$aws/things/RaspberryPi/shadow/get"
         ]
      },
      {
         "Effect": "Allow",
         "Action": "iot:Receive",
         "Resource": [
            "arn:aws:iot:region:account:topic/$aws/things/RaspberryPi/shadow/update/
accepted",
            "arn:aws:iot:region:account:topic/$aws/things/RaspberryPi/shadow/delete/
accepted",
            "arn:aws:iot:region:account:topic/$aws/things/RaspberryPi/shadow/get/
accepted",
            "arn:aws:iot:region:account:topic/$aws/things/RaspberryPi/shadow/update/
rejected",
            "arn:aws:iot:region:account:topic/$aws/things/RaspberryPi/shadow/delete/
rejected"
         ]
      },
      {
         "Effect": "Allow",
         "Action": "iot:Subscribe",
         "Resource": [
            "arn:aws:iot:region:account:topicfilter/$aws/things/RaspberryPi/shadow/
update/accepted",
            "arn:aws:iot:region:account:topicfilter/$aws/things/RaspberryPi/shadow/
delete/accepted",
            "arn:aws:iot:region:account:topicfilter/$aws/things/RaspberryPi/shadow/get/
accepted",
            "arn:aws:iot:region:account:topicfilter/$aws/things/RaspberryPi/shadow/
update/rejected",
            "arn:aws:iot:region:account:topicfilter/$aws/things/RaspberryPi/shadow/
delete/rejected"
         ]
      },
      {
         "Effect": "Allow",
         "Action": [
            "iot:GetThingShadow",
            "iot:UpdateThingShadow",
            "iot:DeleteThingShadow"
         ],
         "Resource": "arn:aws:iot:region:account:thing/RaspberryPi"

      }
   ]
}
```

5. Escolha Create (Criar).

Etapa 2: Criar a coisa, o certificado e a chave privada do AWS IoT

Crie uma coisa no registro do AWS IoT para representar o Raspberry Pi.

1. No console do AWS IoT, no painel de navegação, selecione Manage (Gerenciar) e Things (Coisas).
2. Se uma caixa de diálogo You don't have any things yet (Você ainda não tem coisas) for exibida, selecione Register a thing (Registrar uma coisa). Caso contrário, escolha Create (Criar).
3. Na página Creating AWS IoT things (Criar coisas para IoT), escolha Create a single thing (Criar uma única coisa).
4. Na página Add your device to the device registry (Adicionar o dispositivo ao registro do dispositivo), insira um nome para o item de IoT (por exemplo, **`RaspberryPi`**) e selecione Next (Próximo). Você não pode alterar o nome de uma coisa depois de criá-la. Para alterar o nome de uma coisa, é necessário criar uma coisa, fornecer o novo nome e, depois, excluir a coisa antiga.
5. Na página Add a certificate for your thing (Adicionar um certificado à sua coisa), escolha Create certificate (Criar certificado).
6. Escolha os links Download para fazer download do certificado, da chave privada e do certificado CA raiz.

   > **Important**
   >
   > Esta é a única vez que você pode fazer download do certificado e da chave privada.

7. Para ativar o certificado, escolhaAtivar. O certificado deve estar ativo para que um dispositivo se conecte aoAWS IoT.
8. Selecione Attach a policy (Anexar uma política).
9. Em Add a policy for your thing (Adicionar uma política à coisa), selecione MoistureSensorPolicy e Register Thing (Registrar coisa).

Etapa 3: Criar um tópico do Amazon SNS e uma assinatura

Crie um tópico do Amazon SNS e uma assinatura.

1. DoAWSConsole SNSNo painel de navegação, selecioneTópicose, depois, escolhaCriar tópico.
2. Insira um nome para o tópico (por exemplo, **`MoistureSensorTopic`**).
3. Insira um nome de exibição para o tópico (por exemplo, **`Moisture Sensor Topic`**). Esse é o nome exibido para o tópico no console do Amazon SNS.
4. Escolha Create topic.
5. Na página de detalhes do tópico do Amazon SNS, selecioneCriar assinatura.
6. Em Protocol (Protocolo), escolha Email.
7. Para Endpoint, insira seu endereço de e-mail.
8. Selecione Create subscription.
9. Abra o cliente de e-mail e procure uma mensagem com o assunto **`MoistureSensorTopic`**. Abra o e-mail e clique no link Confirm subscription (Confirmar assinatura).

   > **Important**
   >
   > Você não receberá nenhum alerta por e-mail deste tópico do Amazon SNS até confirmar a assinatura.

Você deve receber uma mensagem de e-mail com o texto digitado.

Etapa 4: Criar uma regra do AWS IoT para enviar um e-mail

Uma regra do AWS IoT define uma consulta e uma ou mais ações a serem executadas quando uma mensagem é recebida de um dispositivo. O mecanismo de regras do AWS IoT escuta mensagens enviadas por dispositivos e usa os dados nas mensagens para determinar se alguma ação deve ser realizada. Para obter mais informações, consulte Regras para AWS IoT (p. 478).

Neste tutorial, o Raspberry Pi publica mensagens no `aws/things/RaspberryPi/shadow/update`. Este é um tópico MQTT interno usado por dispositivos e pelo serviço Thing Shadow. O Raspberry Pi publica mensagens que têm o seguinte formato:

```
{
    "reported": {
        "moisture" : moisture-reading,
        "temp" : temperature-reading
    }
}
```

Você cria uma consulta que extrai os dados de umidade e temperatura da mensagem recebida. Você também cria uma ação do Amazon SNS que usa os dados e os envia aos assinantes do tópico do Amazon SNS se a leitura de umidade estiver abaixo de um valor limite.

Criar uma regra do Amazon SNS

1. No console da AWS IoT no painel de navegação à esquerda, selecione Act (Ação). Se uma caixa de diálogo You don't have any rules yet (Você ainda não tem regras), selecione Create a rule (Criar uma regra). Caso contrário, escolha Create (Criar).
2. Na página Create a rule (Criar uma regra), insira um nome para a regra (por exemplo, **`MoistureSensorRule`**).
3. Em Description (Descrição), forneça uma breve descrição para essa regra (por exemplo, **`Sends an alert when soil moisture level readings are too low`**).
4. Em Rule query statement (Instrução de consulta da regra), escolha o SQL versão 2016-03-23 e insira a seguinte instrução de consulta SQL do AWS IoT:

   ```
   SELECT * FROM '$aws/things/RaspberryPi/shadow/update/accepted' WHERE
    state.reported.moisture < 400
   ```

   Essa instrução aciona a ação da regra quando a leitura de `moisture` é menor que `400`.

   > **Note**
   >
   > Talvez seja necessário usar um valor diferente. Depois de ter o código em execução no Raspberry Pi, você poderá ver os valores obtidos do sensor tocando no sensor, colocando-o na água ou colocando-o em um vaso.
5. Em Set one or more actions (Definir uma ou mais ações), selecione Add action (Adicionar ação).
6. Na página Select an action (Selecionar uma ação), selecione Send a message as an SNS push notification (Enviar uma mensagem como uma notificação por push do SNS).
7. Role até o final da página e selecione Configure action (Configurar ação).
8. Na página Configure action (Configurar ação), em SNS target (Destino do SNS), selecione Select (Selecionar) e LowMoistureTopic.
9. Para Formato de mensagem, selecione RAW.
10. UNDEREscolha ou crie uma função para concederAWS IoTacesso para executar essa ação, escolhaCriar função. Insira um nome para a função (por exemplo, **`LowMoistureTopicRole`**) e escolha Create role (Criar função).

11. Selecione Add action.
12. Selecione Create rule (Criar regra).

Configuração do Raspberry Pi e do sensor de umidade

Insira o cartão microSD no Raspberry Pi, conecte o monitor, o teclado, o mouse e, se você não estiver usando a rede Wi-Fi, o cabo Ethernet. Não conecte o cabo de alimentação ainda.

Conecte o cabo jumper JST ao sensor de umidade. O outro lado do jumper tem quatro cabos:

- Green (Verde): I2C SCL
- Branco: I2C SDA
- Vermelho: alimentação (3,5 V)
- Preto: terra

Mantenha o Raspberry Pi com o conector Ethernet à direita. Nessa orientação, há duas linhas de pinos GPIO na parte superior. Conecte os cabos do sensor de umidade à linha inferior de pinos na ordem a seguir. A partir do pino mais à esquerda, conecte o vermelho (alimentação), o branco (SDA) e o verde (SCL). Ignore um pino e conecte o fio preto (terra). Para obter mais informações, consulte Python Computer Wiring.

Conecte o cabo de alimentação ao Raspberry Pi e conecte a outra extremidade a uma tomada para ligá-lo.

Configurar o Raspberry Pi

1. Em Welcome to Raspberry Pi (Bem-vindo ao Raspberry Pi), selecione Next (Próximo).
2. Escolha o país, o idioma, o fuso horário e o layout do teclado. Escolha Next (Próximo).
3. Insira uma senha para o Raspberry Pi e selecione Next (Próximo).
4. Escolha a rede Wi-Fi e selecione Next (Próximo). Se você não estiver usando uma rede Wi-Fi, selecione Skip (Ignorar).
5. Selecione Next (Próximo) para verificar se há atualizações de software. Quando as atualizações forem concluídas, selecione Restart (Reiniciar) para reiniciar o Raspberry Pi.

Depois que o Raspberry Pi for iniciado, habilite a interface I2C.

1. No canto superior esquerdo da área de trabalho do Raspbian, clique no ícone do Raspberry, selecione Preferences (Preferências) e Raspberry Pi Configuration (Configuração do Raspberry Pi).
2. Na guia Interfaces para I2C, selecione Enable (Habilitar).
3. Escolha OK.

As bibliotecas do sensor de umidade Adafruit STEMMA são escritas para o CircuitPython. Para executá-las em um Raspberry Pi, é necessário instalar a versão mais recente do Python 3.

1. Execute os seguintes comandos em um prompt de comando para atualizar o software do Raspberry Pi:

   ```
   sudo apt-get update
   ```

   ```
   sudo apt-get upgrade
   ```

2. Execute o seguinte comando para atualizar a instalação do Python 3:

   ```
   sudo pip3 install --upgrade setuptools
   ```

3. Execute o seguinte comando para instalar as bibliotecas GPIO do Raspberry Pi:

   ```
   pip3 install RPI.GPIO
   ```

4. Execute o seguinte comando para instalar as bibliotecas Adafruit Blinka:

   ```
   pip3 install adafruit-blinka
   ```

   Para obter mais informações, consulte Instalação de bibliotecas CircuitPython no Raspberry Pi.

5. Execute o seguinte comando para instalar as bibliotecas Adafruit Seesaw:

   ```
   sudo pip3 install adafruit-circuitpython-seesaw
   ```

6. Execute o seguinte comando para instalar o SDK de dispositivo do AWS IoT para Python:

   ```
   pip3 install AWSIoTPythonSDK
   ```

Agora o Raspberry Pi tem todas as bibliotecas necessárias. Crie um arquivo chamado **`moistureSensor.py`** e copie o seguinte código Python no arquivo:

```
from adafruit_seesaw.seesaw import Seesaw
from AWSIoTPythonSDK.MQTTLib import AWSIoTMQTTShadowClient
from board import SCL, SDA

import logging
import time
import json
import argparse
import busio

# Shadow JSON schema:
#
# {
#   "state": {
#       "desired":{
#           "moisture":<INT VALUE>,
#           "temp":<INT VALUE>
#       }
#   }
# }

# Function called when a shadow is updated
def customShadowCallback_Update(payload, responseStatus, token):

    # Display status and data from update request
    if responseStatus == "timeout":
        print("Update request " + token + " time out!")

    if responseStatus == "accepted":
        payloadDict = json.loads(payload)
        print("~~~~~~~~~~~~~~~~~~~~~~~~")
        print("Update request with token: " + token + " accepted!")
        print("moisture: " + str(payloadDict["state"]["reported"]["moisture"]))
        print("temperature: " + str(payloadDict["state"]["reported"]["temp"]))
        print("~~~~~~~~~~~~~~~~~~~~~~~~\n\n")

    if responseStatus == "rejected":
        print("Update request " + token + " rejected!")

# Function called when a shadow is deleted
def customShadowCallback_Delete(payload, responseStatus, token):

     # Display status and data from delete request
```

```
    if responseStatus == "timeout":
        print("Delete request " + token + " time out!")

    if responseStatus == "accepted":
        print("~~~~~~~~~~~~~~~~~~~~~~~~")
        print("Delete request with token: " + token + " accepted!")
        print("~~~~~~~~~~~~~~~~~~~~~~~~\n\n")

    if responseStatus == "rejected":
        print("Delete request " + token + " rejected!")


# Read in command-line parameters
def parseArgs():

    parser = argparse.ArgumentParser()
    parser.add_argument("-e", "--endpoint", action="store", required=True, dest="host",
 help="Your device data endpoint")
    parser.add_argument("-r", "--rootCA", action="store", required=True, dest="rootCAPath",
 help="Root CA file path")
    parser.add_argument("-c", "--cert", action="store", dest="certificatePath",
 help="Certificate file path")
    parser.add_argument("-k", "--key", action="store", dest="privateKeyPath", help="Private
 key file path")
    parser.add_argument("-p", "--port", action="store", dest="port", type=int, help="Port
 number override")
    parser.add_argument("-n", "--thingName", action="store", dest="thingName",
 default="Bot", help="Targeted thing name")
    parser.add_argument("-id", "--clientId", action="store", dest="clientId",
 default="basicShadowUpdater", help="Targeted client id")

    args = parser.parse_args()
    return args


# Configure logging
# AWSIoTMQTTShadowClient writes data to the log
def configureLogging():

    logger = logging.getLogger("AWSIoTPythonSDK.core")
    logger.setLevel(logging.DEBUG)
    streamHandler = logging.StreamHandler()
    formatter = logging.Formatter('%(asctime)s - %(name)s - %(levelname)s - %(message)s')
    streamHandler.setFormatter(formatter)
    logger.addHandler(streamHandler)


# Parse command line arguments
args = parseArgs()

if not args.certificatePath or not args.privateKeyPath:
    parser.error("Missing credentials for authentication.")
    exit(2)

# If no --port argument is passed, default to 8883
if not args.port:
    args.port = 8883


# Init AWSIoTMQTTShadowClient
myAWSIoTMQTTShadowClient = None
myAWSIoTMQTTShadowClient = AWSIoTMQTTShadowClient(args.clientId)
myAWSIoTMQTTShadowClient.configureEndpoint(args.host, args.port)
myAWSIoTMQTTShadowClient.configureCredentials(args.rootCAPath, args.privateKeyPath,
 args.certificatePath)
```

```
# AWSIoTMQTTShadowClient connection configuration
myAWSIoTMQTTShadowClient.configureAutoReconnectBackoffTime(1, 32, 20)
myAWSIoTMQTTShadowClient.configureConnectDisconnectTimeout(10) # 10 sec
myAWSIoTMQTTShadowClient.configureMQTTOperationTimeout(5) # 5 sec

# Initialize Raspberry Pi's I2C interface
i2c_bus = busio.I2C(SCL, SDA)

# Intialize SeeSaw, Adafruit's Circuit Python library
ss = Seesaw(i2c_bus, addr=0x36)

# Connect to AWS IoT
myAWSIoTMQTTShadowClient.connect()

# Create a device shadow handler, use this to update and delete shadow document
deviceShadowHandler = myAWSIoTMQTTShadowClient.createShadowHandlerWithName(args.thingName,
 True)

# Delete current shadow JSON doc
deviceShadowHandler.shadowDelete(customShadowCallback_Delete, 5)

# Read data from moisture sensor and update shadow
while True:

    # read moisture level through capacitive touch pad
    moistureLevel = ss.moisture_read()

    # read temperature from the temperature sensor
    temp = ss.get_temp()

    # Display moisture and temp readings
    print("Moisture Level: {}".format(moistureLevel))
    print("Temperature: {}".format(temp))

    # Create message payload
    payload = {"state":{"reported":{"moisture":str(moistureLevel),"temp":str(temp)}}}

    # Update shadow
    deviceShadowHandler.shadowUpdate(json.dumps(payload), customShadowCallback_Update, 5)
    time.sleep(1)
```

Salve o arquivo em um local onde você possa encontrá-lo. Execute `moistureSensor.py` na linha de comando com os seguintes parâmetros:

endpoint

O endpoint personalizado do AWS IoT Para obter mais informações, consulte API REST do Device Shadow (p. 659).

rootCA

O caminho completo para oAWS IoTCertificado CA raiz.

cert

O caminho completo para o certificado de dispositivo do AWS IoT.

chave

O caminho completo para a chave privada do certificado de dispositivo do AWS IoT.

thingName

O nome da coisa (neste caso, `RaspberryPi`).

clientId

O ID do cliente MQTT. Use `RaspberryPi`.

A linha de comando deve ser semelhante a esta:

```
python3 moistureSensor.py --endpoint your-endpoint --rootCA ~/certs/
AmazonRootCA1.pem --cert ~/certs/raspberrypi-certificate.pem.crt --key ~/certs/
raspberrypi-private.pem.key --thingName RaspberryPi --clientId RaspberryPi
```

Tente tocar no sensor, colocá-lo em um vaso ou colocá-lo em um copo de água para ver como o sensor responde a vários níveis de umidade. Se for necessário, você poderá alterar o valor limite no `MoistureSensorRule`. Quando a leitura do sensor de umidade ficar abaixo do valor especificado na instrução de consulta SQL da regra,AWS IoTPublicará uma mensagem no tópico do Amazon SNS. Você deve receber uma mensagem de e-mail que contém os dados de umidade e temperatura.

Depois de verificar o recebimento de mensagens de e-mail do Amazon SNS, pressioneCTRL+CPara interromper o programa Python. É improvável que o programa Python envie mensagens suficientes para gerar cobranças, mas é uma melhor prática interromper o programa ao concluir.

Gerenciamento de dispositivos com o AWS IoT

A AWS IoT fornece um registro de coisas que ajuda você a gerenciar as coisas. Coisa é uma representação de um dispositivo específico ou entidade lógica. Ela pode ser um dispositivo físico ou sensor (por exemplo, uma lâmpada ou um interruptor em uma parede). Ela também pode ser uma entidade lógica, como uma instância de um aplicativo, ou uma entidade física que não se conecte à AWS IoT, mas esteja relacionada a outros dispositivos que se conectam (por exemplo, um carro que tenha sensores do motor ou um painel de controle).

As informações sobre uma coisa estão armazenadas no registro como dados JSON. Veja um exemplo de coisa:

```
{
     "version": 3,
    "thingName": "MyLightBulb",
    "defaultClientId": "MyLightBulb",
    "thingTypeName": "LightBulb",
    "attributes": {
        "model": "123",
        "wattage": "75"
    }
}
```

As coisas são identificados por um nome. As coisas também podem ter atributos, que são pares de nome-valor que você pode usar para armazenar informações sobre a coisa, como seu número de série ou fabricante.

Um caso típico de uso de dispositivos envolve o uso do nome da coisa como o ID de cliente MQTT padrão. Embora não seja feito um mapeamento entre o nome do registro de uma coisa e seu uso de IDs de cliente MQTT, certificados ou estados do shadow, recomendamos que você escolha um nome de coisa e use-o como o ID de cliente MQTT tanto para o registro quanto para o serviço Device Shadow. Isso proporciona organização e conveniência para sua frota de IoT sem remover a flexibilidade do modelo de certificado do dispositivo ou shadows subjacente.

Não é necessário criar uma coisa no registro para conectar um dispositivo aoAWS IoT. Adicionar coisas ao registro permite que você gerencie e faça pesquisa por dispositivos com mais facilidade.

Como gerenciar coisas com o registro

Você usa oAWS IoTConsole do,AWS IoTAPI ou oAWS CLIPara interagir com o registro. As seções a seguir mostram como usar a CLI para trabalhar com o registro.

Ao nomear objetos da sua coisa:

- Você não deve usar informações pessoalmente identificáveis nos nomes das suas coisas. O nome da coisa pode aparecer em comunicações e relatórios não criptografados.
- Você não deve usar um caractere de dois pontos (:) em um nome de coisa. O caractere de dois pontos é usado como delimitador por outrosAWS IoTserviços e isso pode fazer com que eles analisem strings com nomes de coisas incorretamente.

Criar uma coisa

O comando a seguir mostra como usar o comando AWS IoT da CreateThing na CLI para criar uma coisa. Você não pode mudar o nome de uma coisa depois de criá-la. Para alterar o nome de uma coisa, é necessário criar uma coisa, fornecer o novo nome e, depois, excluir a coisa antiga.

```
$ aws iot create-thing --thing-name "MyLightBulb" --attribute-payload "{\"attributes\":
 {\"wattage\":\"75\", \"model\":\"123\"}}"
```

O comando CreateThing exibe o nome e o nome de recurso da Amazon (ARN) da nova coisa:

```
{
    "thingArn": "arn:aws:iot:us-east-1:123456789012:thing/MyLightBulb",
    "thingName": "MyLightBulb",
    "thingId": "12345678abcdefgh12345678ijklmnop12345678"
}
```

> **Note**
>
> Não recomendamos o uso de informações pessoalmente identificáveis nos nomes das suas coisas.

Listar coisas

Você pode usar o comando ListThings para listar todas as coisas em sua conta:

```
$ aws iot list-things
```

```
{
    "things": [
       {
            "attributes": {
                "model": "123",
                "wattage": "75"
            },
            "version": 1,
            "thingName": "MyLightBulb"
        },
        {
            "attributes": {
                "numOfStates":"3"
             },
            "version": 11,
            "thingName": "MyWallSwitch"
        }
    ]
}
```

Você pode usar oListThingsPara pesquisar todas as coisas de um tipo de coisa específico:

```
$  aws iot list-things --thing-type-name "LightBulb"
```

```
{
    "things": [
        {
            "thingTypeName": "LightBulb",
```

```
            "attributes": {
                "model": "123",
                "wattage": "75"
            },
            "version": 1,
            "thingName": "MyRGBLight"
        },
        {
            "thingTypeName": "LightBulb",
            "attributes": {
                "model": "123",
                "wattage": "75"
            },
            "version": 1,
            "thingName": "MySecondLightBulb"
        }
    ]
}
```

Você pode usar oListThingsPara pesquisar todas as coisas que tenham um atributo com um valor específico. Esse comando pesquisa somente atributos pesquisáveis.

```
$  aws iot list-things --attribute-name "wattage" --attribute-value "75"
```

```
{
    "things": [
        {
            "thingTypeName": "StopLight",
            "attributes": {
                "model": "123",
                "wattage": "75"
            },
            "version": 3,
            "thingName": "MyLightBulb"
        },
        {
            "thingTypeName": "LightBulb",
            "attributes": {
                "model": "123",
                "wattage": "75"
            },
            "version": 1,
            "thingName": "MyRGBLight"
        },
        {
            "thingTypeName": "LightBulb",
            "attributes": {
                "model": "123",
                "wattage": "75"
            },
            "version": 1,
            "thingName": "MySecondLightBulb"
        }
    ]
}
```

Seindexação de frotas de (p. 836)está habilitado, você pode usar osearch-indexpara pesquisar atributos de coisas pesquisáveis e não pesquisáveis, valores de sombra de dispositivo e valores de conectividade. Para obter mais informações sobre o que você pode consultar usando osearch-indexcomando, consulteExemplo de consultas de coisas (p. 858)e a referência da CLI nosearch-indexcomando.

Descreva coisas

Você pode usar oDescribeThingPara exibir informações mais detalhadas sobre uma coisa:

```
$ aws iot describe-thing --thing-name "MyLightBulb"
{
    "version": 3,
    "thingName": "MyLightBulb",
    "thingArn": "arn:aws:iot:us-east-1:123456789012:thing/MyLightBulb",
    "thingId": "12345678abcdefgh12345678ijklmnop12345678",
    "defaultClientId": "MyLightBulb",
    "thingTypeName": "StopLight",
    "attributes": {
        "model": "123",
        "wattage": "75"
    }
}
```

Atualizar uma coisa

Você pode usar o comando UpdateThing para atualizar uma coisa. Observe que este comando atualiza apenas os atributos da coisa. Você não pode mudar o nome de uma coisa. Para alterar o nome de uma coisa, é necessário criar uma coisa, fornecer o novo nome e, depois, excluir a coisa antiga.

```
$ aws iot update-thing --thing-name "MyLightBulb" --attribute-payload "{\"attributes\":
 {\"wattage\":\"150\", \"model\":\"456\"}}"
```

O comando UpdateThing não produz saída. Você pode usar o comando DescribeThing para exibir o resultado:

```
$ aws iot describe-thing --thing-name "MyLightBulb"
{
    "attributes": {
        "model": "456",
        "wattage": "150"
    },
    "version": 2,
    "thingName": "MyLightBulb"
}
```

Excluir uma coisa

Você pode usar o comando DeleteThing para excluir uma coisa:

```
$ aws iot delete-thing --thing-name "MyThing"
```

Esse comando retornará com êxito sem erros se a exclusão for bem-sucedida ou se você especificar algo que não existe.

Anexar uma principal a uma coisa

Um dispositivo físico deve ter um certificado X.509 para se comunicar com a AWS IoT. Você pode associar o certificado em seu dispositivo com a coisa no registro que representa o dispositivo. Para anexar um certificado à coisa, use o comando AttachThingPrincipal:

```
$ aws iot attach-thing-principal --thing-name "MyLightBulb" --principal "arn:aws:iot:us-
east-1:123456789012:cert/a0c01f5835079de0a7514643d68ef8414ab739a1e94ee4162977b02b12842847"
```

O comando AttachThingPrincipal não produz nenhuma saída.

Desanexar uma principal de uma coisa

Você pode usar o comando DetachThingPrincipal para desanexar um certificado de uma coisa:

```
$ aws iot detach-thing-principal --thing-name "MyLightBulb" --principal "arn:aws:iot:us-
east-1:123456789012:cert/a0c01f5835079de0a7514643d68ef8414ab739a1e94ee4162977b02b12842847"
```

ODetachThingPrincipalO comando não produz nenhuma saída.

Tipos de coisa

Os tipos de coisa permitem que você armazene descrições e informações de configuração que sejam comuns a todas as coisas associadas com o mesmo tipo. Isso simplifica o gerenciamento de coisas no registro. Por exemplo, você pode definir um tipo de coisa LightBulb (Lâmpada). Todas as coisas associadas ao tipo de coisa LightBulb compartilham um conjunto de atributos: número de série, fabricante e potência. Quando você cria uma coisa do tipo LightBulb (ou altera o tipo de uma coisa existente para LightBulb), é possível especificar valores para cada um dos atributos definidos no tipo de coisa LightBulb.

Embora os tipos de coisa sejam opcionais, seu uso facilita a descobrir coisas.

- As coisas com um tipo de coisa podem ter até 50 atributos.
- As coisas sem um tipo de coisa podem ter até três atributos.
- Uma coisa só pode ser associada a um tipo de coisa.
- Não há limite para o número de tipos de coisa que você pode criar em sua conta.

Os tipos de coisa são imutáveis. Não é possível alterar um nome de tipo de coisa após sua criação. Você pode reprovar um tipo de coisa a qualquer momento para impedir que novas coisas sejam associadas a ele. Você também pode excluir tipos de coisa que não tenham coisas associadas a eles.

Criar um tipo de coisa

Você pode usar o comando CreateThingType para criar um tipo de coisa:

```
$ aws iot create-thing-type

                --thing-type-name "LightBulb" --thing-type-properties
 "thingTypeDescription=light bulb type, searchableAttributes=wattage,model"
```

O comando CreateThingType retorna uma resposta que contém o tipo de coisa e seu ARN:

```
{
    "thingTypeName": "LightBulb",
    "thingTypeId": "df9c2d8c-894d-46a9-8192-9068d01b2886",
    "thingTypeArn": "arn:aws:iot:us-west-2:123456789012:thingtype/LightBulb"
```

```
}
```

Listar tipos de coisa

Você pode usar o comando ListThingTypes para listar tipos de coisa:

```
$ aws iot list-thing-types
```

OListThingTypesO comando retorna uma lista dos tipos de coisa definidos em seuConta da AWS:

```
{
    "thingTypes": [
        {
            "thingTypeName": "LightBulb",
            "thingTypeProperties": {
                "searchableAttributes": [
                    "wattage",
                    "model"
                ],
                "thingTypeDescription": "light bulb type"
            },
            "thingTypeMetadata": {
                "deprecated": false,
                "creationDate": 1468423800950
            }
        }
    ]
}
```

Descrever um tipo de coisa

Você pode usar o comando DescribeThingType para obter informações sobre um tipo de coisa:

```
$ aws iot describe-thing-type --thing-type-name "LightBulb"
```

O comando DescribeThingType retorna informações sobre o tipo especificado:

```
{
    "thingTypeProperties": {
        "searchableAttributes": [
            "model",
            "wattage"
        ],
        "thingTypeDescription": "light bulb type"
    },
    "thingTypeId": "df9c2d8c-894d-46a9-8192-9068d01b2886",
    "thingTypeArn": "arn:aws:iot:us-west-2:123456789012:thingtype/LightBulb",
    "thingTypeName": "LightBulb",
    "thingTypeMetadata": {
        "deprecated": false,
        "creationDate": 1544466338.399
    }
}
```

Associar um tipo de coisa a uma coisa

Você pode usar o comando CreateThing para especificar um tipo de coisa ao criar uma coisa:

```
$ aws iot create-thing --thing-name "MyLightBulb" --thing-type-name "LightBulb" --
attribute-payload "{\"attributes\": {\"wattage\":\"75\", \"model\":\"123\"}}"
```

Você pode usar o comando UpdateThing a qualquer momento para alterar o tipo de coisa associado a uma coisa:

```
$ aws iot update-thing --thing-name "MyLightBulb"
                --thing-type-name "LightBulb" --attribute-payload  "{\"attributes\":
 {\"wattage\":\"75\", \"model\":\"123\"}}"
```

Você também pode usar o comando UpdateThing para desassociar uma coisa de um tipo de coisa.

Reprovar um tipo de coisa

Os tipos de coisa são imutáveis. Eles não podem ser alterados depois que são definidos. No entanto, você pode reprovar um tipo de coisa para impedir que os usuários associem novas coisas a ele. Todas as coisas existentes associadas ao tipo de coisa são inalteradas.

Para obsoletar um tipo de coisa, use o comando DeprecateThingType:

```
$ aws iot deprecate-thing-type --thing-type-name "myThingType"
```

Você pode usar o comando DescribeThingType para exibir o resultado:

```
$ aws iot describe-thing-type --thing-type-name "StopLight":
```

```
{
    "thingTypeName": "StopLight",
    "thingTypeProperties": {
        "searchableAttributes": [
            "wattage",
            "numOfLights",
            "model"
        ],
        "thingTypeDescription": "traffic light type",
    },
    "thingTypeMetadata": {
        "deprecated": true,
        "creationDate": 1468425854308,
        "deprecationDate": 1468446026349
    }
}
```

Reprovar um tipo de coisa é uma operação reversível. Você pode desfazer uma reprovação usando o sinalizador `--undo-deprecate` com o comando CLI DeprecateThingType:

```
$ aws iot deprecate-thing-type --thing-type-name "myThingType" --undo-deprecate
```

Você pode usar o comando CLI DescribeThingType para exibir o resultado:

```
$ aws iot describe-thing-type --thing-type-name "StopLight":
```

```
{
    "thingTypeName": "StopLight",
```

```
    "thingTypeArn": "arn:aws:iot:us-east-1:123456789012:thingtype/StopLight",
    "thingTypeId": "12345678abcdefgh12345678ijklmnop12345678"
    "thingTypeProperties": {
        "searchableAttributes": [
            "wattage",
            "numOfLights",
            "model"
        ],
        "thingTypeDescription": "traffic light type"
    },
    "thingTypeMetadata": {
        "deprecated": false,
        "creationDate": 1468425854308,
    }
}
```

Excluir um tipo de coisa

É possível excluir tipos de coisa somente depois que eles forem reprovados. Para excluir um tipo de coisa, use o comando DeleteThingType:

```
$ aws iot delete-thing-type --thing-type-name "StopLight"
```

> **Note**
>
> É necessário esperar cinco minutos depois de reprovar um tipo de coisa antes que possa excluí-lo.

Grupos de coisas estáticas

Os grupos de coisas estáticas permitem gerenciar várias coisas simultaneamente, categorizando-as em grupos. Os grupos de coisas estáticas contêm um grupo de coisas que são gerenciadas usando o console, a CLI ou a API. Grupos de coisas dinâmicas (p. 282), por outro lado, contêm coisas que correspondem a uma consulta especificada. Os grupos de coisas estáticos também podem conter outros grupos de coisas estáticos — você pode criar uma hierarquia de grupos. Você pode anexar uma política a um grupo pai, e ela será herdada pelos grupos filho, e por todas as coisas do grupo e também dos grupos filho. Isso facilita o controle de permissões para grandes números de coisas.

Veja o que você pode fazer com grupos de coisas estáticas:

- Criar, descrever ou excluir um grupo.
- Adicionar uma coisa a um grupo ou a mais de um grupo.
- Remover uma coisa de um grupo.
- Listar os grupos que você criou.
- Listar todos os grupos filho de um grupo (seus descendentes diretos e indiretos.)
- Listar as coisas em um grupo, incluindo todas as coisas em seus grupos filho.
- Listar todos os grupos ancestrais de um grupo (seus pais diretos e indiretos.)
- Adicionar, excluir ou atualizar os atributos de um grupo. Os atributos são pares nome-valor que você pode usar para armazenar informações sobre um grupo.
- Anexar ou desanexar uma política de ou para um grupo.
- Listar as políticas anexadas a um grupo.
- Listar as políticas herdadas por uma coisa (em virtude das políticas anexadas a seu grupo ou um de seus grupos pais.)

- Configure as opções de registro em log para as coisas em um grupo. Consulte Configurar registro em log da AWS IoT (p. 433).
- Crie trabalhos que serão enviados para e executados em cada coisa em um grupo e nos respectivos grupos filho. Consulte Trabalhos (p. 681).

Estas são algumas limitações de grupos de coisas estáticas:

- Um grupo pode ter no máximo um pai direto.
- Se um grupo for filho de outro grupo, você deve especificar isso no momento em que for criado.
- Você não pode alterar o pai de um grupo mais tarde, portanto, certifique-se de planejar sua hierarquia de grupos e criar um grupo pai antes de qualquer grupo filho que ele contenha.
- O número de grupos aos quais uma coisa pode pertencer é limitado.
- Não é possível adicionar uma coisa a mais de um grupo na mesma hierarquia. Ou seja, não é possível adicionar uma coisa a dois grupos que compartilham o mesmo pai.
- Não é possível renomear um grupo.
- Nomes de coisas não podem conter caracteres internacionais, como, û, é e ñ.
- Você não deve usar informações pessoalmente identificáveis nos nomes dos grupos de coisas. O nome do grupo de coisas pode aparecer em comunicações e relatórios não criptografados.
- Você não deve usar um caractere de dois pontos (:) em um nome de grupo de coisas. O caractere de dois pontos é usado como delimitador por outrosAWS IoTserviços e isso pode fazer com que eles analisem strings com nomes de grupos de coisas incorretamente.

Anexar e desanexar políticas em grupos pode aprimorar a segurança das operações da AWS IoT de várias maneiras significativas. O método por dispositivo de anexar uma política a um certificado, que, depois, é anexado a uma coisa, é demorado e dificulta a atualização ou a alteração rápida de políticas em toda a frota de dispositivos. Ter uma política anexada ao grupo da coisa economiza etapas na hora de alternar os certificados em uma coisa. Além disso, as políticas são aplicadas dinamicamente a coisas quando suas associações a grupos é alterada, portanto, não é necessário recriar um conjunto complexo de permissões toda vez que a associação de um dispositivo é alterada em um grupo.

Criar um grupo de coisas estáticas

Use o comando CreateThingGroup para criar um grupo de coisas estáticas:

```
$ aws iot create-thing-group --thing-group-name LightBulbs
```

O comando CreateThingGroup retorna uma resposta que contém o nome, ID e ARN do grupo de coisas estáticas:

```
{
    "thingGroupName": "LightBulbs",
    "thingGroupId": "abcdefgh12345678ijklmnop12345678qrstuvwx",
    "thingGroupArn": "arn:aws:iot:us-west-2:123456789012:thinggroup/LightBulbs"
}
```

> **Note**
>
> Não recomendamos o uso de informações pessoalmente identificáveis nos nomes dos grupos de coisas.

Este é um exemplo que especifica um pai do grupo de coisas estáticas quando ele é criado:

```
$ aws iot create-thing-group --thing-group-name RedLights --parent-group-name LightBulbs
```

Como anteriormente, o comando CreateThingGroup retorna uma resposta que contém o grupo de coisas, seu ID e ARN:

```
{
    "thingGroupName": "RedLights",
    "thingGroupId": "abcdefgh12345678ijklmnop12345678qrstuvwx",
    "thingGroupArn": "arn:aws:iot:us-west-2:123456789012:thinggroup/RedLights",
}
```

> **Important**
>
> Lembre-se dos seguintes limites ao criar as hierarquias de grupo de coisas:
>
> - Um grupo de coisas pode ter apenas um pai direto.
> - O número de grupos filhos diretos que um grupo pode ter é limitado.
> - A profundidade máxima de uma hierarquia de grupos é limitada.
> - O número de atributos que um grupo de coisas pode ter é limitado. Os atributos são pares nome-valor que você pode usar para armazenar informações sobre um grupo. Os comprimentos de cada nome de atributo e cada valor também são limitados.

Descrever um grupo de coisas

Você pode usar o comando DescribeThingGroup para obter informações sobre um grupo de coisas:

```
$ aws iot describe-thing-group --thing-group-name RedLights
```

O comando DescribeThingGroup retorna informações sobre o grupo especificado:

```
{
    "thingGroupName": "RedLights",
    "thingGroupArn": "arn:aws:iot:us-west-2:123456789012:thinggroup/RedLights",
    "thingGroupId": "12345678abcdefgh12345678ijklmnop12345678",
    "version": 1,
    "thingGroupMetadata": {
        "creationDate": 1478299948.882
        "parentGroupName": "Lights",
        "rootToParentThingGroups": [
            {
                "groupArn": "arn:aws:iot:us-west-2:123456789012:thinggroup/ShinyObjects",
                "groupName": "ShinyObjects"
            },
            {
                "groupArn": "arn:aws:iot:us-west-2:123456789012:thinggroup/LightBulbs",
                "groupName": "LightBulbs"
            }
        ]
    },
    "thingGroupProperties": {
        "attributePayload": {
            "attributes": {
                "brightness": "3400_lumens"
            },
        },
        "thingGroupDescription": "string"
```

```
    },
}
```

Adicionar uma coisa a um grupo de coisas estáticas

Você pode usar o comando AddThingToThingGroup para adicionar uma coisa a um grupo de coisas estáticas:

```
$ aws iot add-thing-to-thing-group --thing-name MyLightBulb --thing-group-name RedLights
```

O comando AddThingToThingGroup não produz nenhuma saída.

> **Important**
>
> Você pode adicionar uma coisa a um máximo de 10 grupos. Mas você não pode adicionar uma coisa a mais de um grupo na mesma hierarquia. Ou seja, não é possível adicionar uma coisa a dois grupos que compartilham o mesmo pai.
> Se uma coisa pertencer ao máximo de grupos de coisas possível, e um ou mais desses grupos for um grupo de coisas dinâmicas, você poderá usar o sinalizador overrideDynamicGroups para fazer com que grupos estáticos tenham prioridade sobre grupos dinâmicos.

Remover uma coisa de um grupo de coisas estáticas

Você pode usar o comando RemoveThingFromThingGroup para remover uma coisa de um grupo:

```
$ aws iot remove-thing-from-thing-group --thing-name MyLightBulb --thing-group-name
 RedLights
```

O comando RemoveThingFromThingGroup não produz nenhuma saída.

Listar coisas em um grupo de coisas

Você pode usar o comando ListThingsInThingGroup para listar coisas que pertencem a um grupo:

```
$ aws iot list-things-in-thing-group --thing-group-name LightBulbs
```

O comando ListThingsInThingGroup retorna uma lista das coisas no grupo especificado:

```
{
    "things":[
        "TestThingA"
    ]
}
```

Com o parâmetro `--recursive`, você pode listar coisas que pertencem a um grupo e a qualquer um dos grupos filho:

```
$ aws iot list-things-in-thing-group --thing-group-name LightBulbs --recursive
```

```
{
    "things":[
        "TestThingA",
```

```
        "MyLightBulb"
    ]
}
```

Note

Essa operação é eventualmente consistente. Em outras palavras, alterações no grupo de coisas podem não ser refletidas imediatamente.

Listar grupos de coisas

Você pode usar o comando ListThingGroups para listar grupos de coisas da sua conta:

```
$ aws iot list-thing-groups
```

OListThingGroupsO comando retorna uma lista dos grupos de coisas em seuConta da AWS:

```
{
    "thingGroups": [
        {
            "groupName": "LightBulbs",
            "groupArn": "arn:aws:iot:us-west-2:123456789012:thinggroup/LightBulbs"
        },
        {
            "groupName": "RedLights",
            "groupArn": "arn:aws:iot:us-west-2:123456789012:thinggroup/RedLights"
        },
        {
            "groupName": "RedLEDLights",
            "groupArn": "arn:aws:iot:us-west-2:123456789012:thinggroup/RedLEDLights"
        },
        {
            "groupName": "RedIncandescentLights",
            "groupArn": "arn:aws:iot:us-west-2:123456789012:thinggroup/
RedIncandescentLights"
        }
        {
            "groupName": "ReplaceableObjects",
            "groupArn": "arn:aws:iot:us-west-2:123456789012:thinggroup/ReplaceableObjects"
        }
    ]
}
```

Use os filtros opcionais para listar os grupos que têm um determinado grupo como pai (`--parent-group`) ou grupos cujos nomes começam com um determinado prefixo (`--name-prefix-filter`). O parâmetro `--recursive` permite que você liste todos os grupos filho, não apenas grupos filho diretos de um grupo de coisas:

```
$ aws iot list-thing-groups --parent-group LightBulbs
```

Neste caso, oListThingGroupsO comando retorna uma lista dos grupos filho diretos do grupo de coisas definido em seuConta da AWS:

```
{
    "childGroups":[
        {
            "groupName": "RedLights",
            "groupArn": "arn:aws:iot:us-west-2:123456789012:thinggroup/RedLights"
```

```
        }
    ]
}
```

Use o parâmetro `--recursive` com o comando ListThingGroups para listar todos os grupos filho de um grupo de coisas, não apenas os filhos diretos:

```
$ aws iot list-thing-groups --parent-group LightBulbs --recursive
```

O comando ListThingGroups retorna uma lista de todos os grupos filho de um grupo de coisas:

```
{
    "childGroups":[
        {
            "groupName": "RedLights",
            "groupArn": "arn:aws:iot:us-west-2:123456789012:thinggroup/RedLights"
        },
        {
            "groupName": "RedLEDLights",
            "groupArn": "arn:aws:iot:us-west-2:123456789012:thinggroup/RedLEDLights"
        },
        {
            "groupName": "RedIncandescentLights",
            "groupArn": "arn:aws:iot:us-west-2:123456789012:thinggroup/
RedIncandescentLights"
        }
    ]
}
```

> **Note**
>
> Essa operação é eventualmente consistente. Em outras palavras, alterações no grupo de coisas podem não ser refletidas imediatamente.

Listar grupos para uma coisa

Você pode usar o comando ListThingGroupsForThing para listar os grupos aos quais uma coisa pertence, incluindo todos os grupos pai:

```
$ aws iot list-thing-groups-for-thing --thing-name MyLightBulb
```

O comando ListThingGroupsForThing retorna uma lista dos grupos de coisas aos quais a coisa pertence, incluindo todos os grupos pai:

```
{
    "thingGroups":[
        {
            "groupName": "LightBulbs",
            "groupArn": "arn:aws:iot:us-west-2:123456789012:thinggroup/LightBulbs"
        },
        {
            "groupName": "RedLights",
            "groupArn": "arn:aws:iot:us-west-2:123456789012:thinggroup/RedLights"
        },
        {
            "groupName": "ReplaceableObjects",
            "groupArn": "arn:aws:iot:us-west-2:123456789012:thinggroup/ReplaceableObjects"
        }
```

```
    ]
}
```

Atualizar um grupo de coisas estáticas

Você pode usar o comando UpdateThingGroup para atualizar os atributos de um grupo de coisas estáticas:

```
$ aws iot update-thing-group --thing-group-name "LightBulbs" --thing-group-properties
 "thingGroupDescription=\"this is a test group\", attributePayload=\"{\"attributes
\"={\"Owner\"=\"150\",\"modelNames\"=\"456\"}}"
```

O comando UpdateThingGroup retorna uma resposta que contém o número da versão do grupo depois da atualização:

```
{
    "version": 4
}
```

> **Note**
>
> O número de atributos que uma coisa pode ter é limitado.

Excluir um grupo de coisas

Para excluir um grupo de coisas, use o comando DeleteThingGroup:

```
$ aws iot delete-thing-group --thing-group-name "RedLights"
```

O comando DeleteThingGroup não produz nenhuma saída.

> **Important**
>
> Se você tentar excluir um grupo de coisas que tem grupos de coisas filho, você receberá um erro:
>
> ```
> A client error (InvalidRequestException) occurred when calling the
>  DeleteThingGroup
> operation: Cannot delete thing group : RedLights when there are still child groups
>  attached to it.
> ```
>
> Você deve excluir todos os grupos de coisas filho primeiro antes de excluir o grupo.

Você pode excluir um grupo que tem coisas filho, mas as permissões concedidas para as coisas por associação no grupo não se aplicarão mais. Antes de excluir um grupo que tem uma política anexada, verifique cuidadosamente se a remoção dessas permissões não fará com que as coisas no grupo não funcionem corretamente. Além disso, observe que os comandos que mostram a quais grupos uma coisa pertence (por exemplo, ListGroupsForThing) podem continuar a mostrar o grupo enquanto os registros na nuvem estão sendo atualizados.

Anexar uma política a um grupo de coisas estáticas

Você pode usar o comando AttachPolicy para anexar uma política a um grupo de coisas estáticas e, por extensão, a todas as coisas daquele grupo e coisas em qualquer um dos grupos filho:

```
$ aws iot attach-policy \
  --target "arn:aws:iot:us-west-2:123456789012:thinggroup/LightBulbs" \
  --policy-name "myLightBulbPolicy"
```

O comando AttachPolicy não produz nenhuma saída

> **Important**
>
> Você pode anexar a quantidade máxima de duas políticas a um grupo.
>
> **Note**
>
> Não recomendamos o uso de informações pessoalmente identificáveis nos nomes de política.

O`--target`O parâmetro pode ser o ARN de um grupo de coisas (conforme acima), o ARN de um certificado ou uma identidade do Amazon Cognito. Para obter mais informações sobre políticas, certificados e autenticação, consulte Autenticação (p. 294).

Para obter mais informações, consulteAWS IoT Corepolíticas.

Desanexar uma política de um grupo de coisas estáticas

Você pode usar o comando DetachPolicy para desanexar uma política de um grupo de coisas e, portanto, por extensão, de todas as coisas daquele grupo e coisas em qualquer um dos grupos filho:

```
$ aws iot detach-policy --target "arn:aws:iot:us-west-2:123456789012:thinggroup/LightBulbs"
 --policy-name "myLightBulbPolicy"
```

O comando DetachPolicy não produz nenhuma saída.

Listar as políticas anexadas a um grupo de coisas estáticas

Você pode usar o comando ListAttachedPolicies para listar as políticas anexadas a um grupo de coisas estáticas:

```
$ aws iot list-attached-policies --target "arn:aws:iot:us-west-2:123456789012:thinggroup/
RedLights"
```

O`--target`O parâmetro pode ser o ARN de um grupo de coisas (conforme acima), o ARN de um certificado ou uma identidade do Amazon Cognito.

Adicione o parâmetro `--recursive` opcional para incluir todas as políticas anexadas aos grupos pai do grupo.

O comando ListAttachedPolicies retorna uma lista de políticas:

```
{
    "policies": [
        "MyLightBulbPolicy"
        ...
    ]
```

```
}
```

Listar os grupos para uma política

Você pode usar o comando ListTargetsForPolicy para listar os destinos, incluindo todos os grupos, aos quais uma política está anexada:

```
$ aws iot list-targets-for-policy --policy-name "MyLightBulbPolicy"
```

Adicione o parâmetro `--page-size` *`number`* opcional para especificar o número máximo de resultados a serem retornados em cada consulta, e o parâmetro `--marker` *`string`* em chamadas subsequentes para recuperar o próximo conjunto de resultados, se houver.

O comando ListTargetsForPolicy retorna uma lista de destinos e o token a ser usado para recuperar mais resultados:

```
{
    "nextMarker": "string",
    "targets": [ "string" ... ]
}
```

Obter políticas efetivas para uma coisa

Você pode usar o comando GetEffectivePolicies para listar as políticas em vigor para uma coisa, incluindo as políticas anexadas a todos os grupos aos quais a coisa pertence (se o grupo for um pai direto ou um ancestral indireto):

```
$ aws iot get-effective-policies \
  --thing-name "MyLightBulb" \
  --principal "arn:aws:iot:us-east-1:123456789012:cert/
a0c01f5835079de0a7514643d68ef8414ab739a1e94ee4162977b02b12842847"
```

Use o parâmetro `--principal` para especificar o ARN do certificado anexado à coisa. Se você estiver usando a autenticação de identidade do Amazon Cognito, use o`--cognito-identity-pool-id`parâmetro e, opcionalmente, adicione o`--principal`Para especificar uma identidade do Amazon Cognito. Se você especificar apenas o `--cognito-identity-pool-id`, as políticas associadas a essa função do grupo de identidades para usuários não autenticados serão retornadas. Se usar os dois, as políticas associadas a essa função do grupo de identidades para usuários autenticados serão retornadas.

O parâmetro `--thing-name` é opcional e pode ser usado em vez do parâmetro `--principal`. Quando o parâmetro é usado, as políticas anexadas a qualquer grupo ao qual a coisa pertence e as políticas anexadas a qualquer grupo pai desses grupos (até o grupo raiz na hierarquia) são retornadas.

O comando GetEffectivePolicies retorna uma lista de políticas:

```
{
    "effectivePolicies": [
        {
            "policyArn": "string",
            "policyDocument": "string",
            "policyName": "string"
        }
        ...
    ]
}
```

Testar a autorização de ações MQTT

Você pode usar oTestAuthorizationcomando para testar se umMQTTação (`Publish`,`Subscribe`) é permitido para uma coisa:

```
aws iot test-authorization \
    --principal "arn:aws:iot:us-east-1:123456789012:cert/
a0c01f5835079de0a7514643d68ef8414ab739a1e94ee4162977b02b12842847" \
    --auth-infos "{\"actionType\": \"PUBLISH\", \"resources\": [ \"arn:aws:iot:us-
east-1:123456789012:topic/my/topic\"]}"
```

Use o parâmetro `--principal` para especificar o ARN do certificado anexado à coisa. Se estiver usando a autenticação do Amazon Cognito, especifique uma identidade do Cognito como a`--principal`ou use o`--cognito-identity-pool-id`parâmetro, ou ambos. Se você especificar apenas o `--cognito-identity-pool-id`, as políticas associadas a essa função do grupo de identidades para usuários não autenticados serão consideradas. Se você usar os dois parâmetros, as políticas associadas a essa função do grupo de identidades para usuários autenticados serão consideradas.

Especifique uma ou mais ações MQTT que você deseja testar listando conjuntos de recursos e tipos de ação depois do parâmetro `--auth-infos`. O campo `actionType` deve conter "PUBLISH", "SUBSCRIBE", "RECEIVE" ou "CONNECT". O campo `resources` deve conter uma lista de ARNs de recursos. Consulte Políticas do AWS IoT Core (p. 332) para obter mais informações.

Você pode testar os efeitos da adição de políticas especificando-as com o parâmetro `--policy-names-to-add`. Ou pode testar os efeitos da remoção de políticas especificando-as com o parâmetro `--policy-names-to-skip`.

Você pode usar o parâmetro `--client-id` opcional para refinar ainda mais os resultados.

O comando TestAuthorization retorna detalhes das ações que foram permitidas ou negadas para cada conjunto de consultas `--auth-infos` que você especificou:

```
{
    "authResults": [
        {
            "allowed": {
                "policies": [
                    {
                        "policyArn": "string",
                        "policyName": "string"
                    }
                ]
            },
            "authDecision": "string",
            "authInfo": {
                "actionType": "string",
                "resources": [ "string" ]
            },
            "denied": {
                "explicitDeny": {
                    "policies": [
                        {
                            "policyArn": "string",
                            "policyName": "string"
                        }
                    ]
                },
                "implicitDeny": {
                    "policies": [
                        {
```

```
                            "policyArn": "string",
                            "policyName": "string"
                        }
                    ]
                }
            },
            "missingContextValues": [ "string" ]
        }
    ]
}
```

Grupos de coisas dinâmicas

> **Note**
>
> O recurso de indexação de frota para suportar indexação denominada sombras eAWS IoT Device DefenderOs dados de violações estão em versão de visualização para oAWS IoTGerenciamento de dispositivos e sujeita a alterações.

Os grupos de coisas dinâmicos atualizam a associação a grupos por meio de consultas de pesquisa. Ao usar grupos de coisas dinâmicos, é possível alterar a maneira como você interage com as coisas em função da sua conectividade, registro, shadow ou dados de violações do Device Defender. Como os grupos de coisas dinâmicos estão vinculados ao seu índice de frota, você deve habilitar a indexação de frota para usá-los. Você pode visualizar as coisas em um grupo dinâmico antes de criar o grupo por meio de uma consulta de pesquisa de indexação de frota. Para obter mais informações, consulte Indexação de frota (p. 836) e Sintaxe de consulta (p. 857).

Você pode especificar um grupo de coisas dinâmico como destino para um trabalho. Somente as coisas que atendem aos critérios que definem o grupo de coisas dinâmico executam o trabalho.

Por exemplo, suponha que você queira atualizar o firmware em seus dispositivos, mas, para minimizar a chance de a atualização ser interrompida, você só deseja atualizar o firmware em dispositivos com duração de bateria maior que 80%. Você pode criar um grupo de coisas dinâmico que inclua apenas dispositivos com duração de bateria relatada acima de 80%, e você pode usar esse grupo de coisas dinâmico como destino para o seu trabalho de atualização de firmware. Somente dispositivos que atendem aos critérios de duração da bateria recebem a atualização do firmware. Conforme os dispositivos atingem os critérios de 80% de duração da bateria, eles são adicionados ao grupo de coisas dinâmico e recebem a atualização do firmware.

Para obter mais informações sobre como especificar grupos de coisas como destinos de trabalhos, consulteCreateJob.

Os grupos de coisas dinâmicos diferem dos grupos de coisas estáticos das seguintes maneiras:

- A associação de coisas não é explicitamente definida. Para criar um grupo dinâmico, você deve definirUma string de consulta (p. 858)que define a associação ao grupo.
- Grupos de coisas dinâmicos não podem fazer parte de uma hierarquia.
- Grupos de coisas dinâmicos não podem ter políticas aplicadas a eles.
- Use um conjunto diferente de comandos para criar, atualizar e excluir grupos de coisas dinâmicos. Para todas as outras operações, os mesmos comandos que você usa para interagir com grupos de coisas estáticos podem ser usados para interagir com grupos de coisas dinâmicos.
- O número de grupos dinâmicos que uma única conta pode ter é limitado.

- Você não deve usar informações pessoalmente identificáveis nos nomes dos grupos de coisas. O nome do grupo de coisas pode aparecer em comunicações e relatórios não criptografados.
- Você não deve usar um caractere de dois pontos (:) em um nome de grupo de coisas. O caractere de dois pontos é usado como delimitador por outrosAWS IoTserviços e isso pode fazer com que eles analisem strings com nomes de grupos de coisas incorretamente.

Para obter mais informações sobre grupos de coisas estáticos, consulte Grupos de coisas estáticas (p. 272).

Como exemplo, suponha que criamos um grupo dinâmico que contém todas as salas em um depósito cuja temperatura é maior que 60 graus Fahrenheit. Quando a temperatura de uma sala é de 61 graus ou mais, ela é adicionada ao grupo de coisas dinâmico RoomTooWarm. Todas as salas do grupo de coisas dinâmico RoomTooWarm teriam ventiladores ligados. Quando a temperatura de uma sala cair para 60 graus ou menos, ela será removida do grupo de coisas dinâmico e seu ventilador será desligado.

Criar grupo de coisas dinâmicas

Use o comando CreateDynamicThingGroup para criar um grupo de coisas dinâmico. Para criar um grupo de coisas dinâmico para o cenário com sala quente demais, use o comando create-dynamic-thing-group da CLI:

```
$ aws iot create-dynamic-thing-group --thing-group-name "RoomTooWarm" --query-string
 "attributes.temperature>60"
```

> **Note**
>
> Não recomendamos o uso de informações pessoalmente identificáveis nos nomes dos grupos de coisas dinâmico.

OCreateDynamicThingGroupO comando retorna uma resposta que contém o nome do índice, a string de consulta, a versão de consulta, o nome do grupo de coisas, o ID do grupo de coisas e o ARN (Amazon Resource Name) do grupo de coisas:

```
{
    "indexName": "AWS_Things",
    "queryVersion": "2017-09-30",
    "thingGroupName": "RoomTooWarm",
    "thingGroupArn": "arn:aws:iot:us-west-2:123456789012:thinggroup/RoomTooWarm",
    "queryString": "attributes.temperature>60\n",
    "thingGroupId": "abcdefgh12345678ijklmnop12345678qrstuvwx"
}
```

A criação do grupo de coisas dinâmico não é instantânea. A alocação do grupo de coisas dinâmico leva tempo para ser concluída. Quando um grupo de coisas dinâmico é criado, o status do grupo é definido como `BUILDING`. Quando a alocação é concluída, o status muda para `ACTIVE`. Para verificar o status do seu grupo de coisas dinâmico, use o comando DescribeThingGroup.

Descrever um grupo de coisas dinâmicas

Use o comando DescribeThingGroup para obter informações sobre um grupo de coisas dinâmico:

```
$ aws iot describe-thing-group --thing-group-name "RoomTooWarm"
```

O comando DescribeThingGroup retorna informações sobre o grupo especificado:

```
{
    "status": "ACTIVE",
    "indexName": "AWS_Things",
    "thingGroupName": "RoomTooWarm",
    "thingGroupArn": "arn:aws:iot:us-west-2:123456789012:thinggroup/RoomTooWarm",
    "queryString": "attributes.temperature>60\n",
    "version": 1,
    "thingGroupMetadata": {
        "creationDate": 1548716921.289
    },
    "thingGroupProperties": {},
    "queryVersion": "2017-09-30",
    "thingGroupId": "84dd9b5b-2b98-4c65-84e4-be0e1ecf4fd8"
}
```

A execução do DescribeThingGroup em um grupo de coisas dinâmico retorna atributos específicos para grupos de coisas dinâmicos, como o queryString e o status.

O status de um grupo de coisas dinâmico pode ter os seguintes valores:

`ACTIVE`

O grupo de coisas dinâmico está pronto para uso.

`BUILDING`

O grupo de coisas dinâmico está sendo criado e a associação de coisas está sendo processada.

`REBUILDING`

A associação do grupo de coisas dinâmico está sendo atualizada, seguindo o ajuste da consulta de pesquisa do grupo.

> **Note**
>
> Depois de criar um grupo de coisas dinâmico, você pode usar o grupo, independentemente do status. Somente grupos de coisas dinâmico com status `ACTIVE` incluem todas as coisas que correspondem à consulta de pesquisa para esse grupo de coisas dinâmico. Grupos de coisas dinâmicos com status `BUILDING` e `REBUILDING` podem não incluir todas as coisas que correspondem à consulta de pesquisa.

Atualizar um grupo de coisas dinâmicas

Use o comando UpdateDynamicThingGroup para atualizar os atributos de um grupo de coisas dinâmico, incluindo a consulta de pesquisa do grupo. O comando a seguir atualiza a descrição do grupo de coisas e a string de consulta que altera os critérios de associação para temperatura > 65:

```
$ aws iot update-dynamic-thing-group --thing-group-name "RoomTooWarm" --thing-group-
properties "thingGroupDescription=\"This thing group contains rooms warmer than 65F.\"" --
query-string "attributes.temperature>65"
```

O comando UpdateDynamicThingGroup retorna uma resposta que contém o número da versão do grupo depois da atualização:

```
{
    "version": 2
}
```

As atualizações do grupo de coisas dinâmico não são instantâneas. A alocação do grupo de coisas dinâmico leva tempo para ser concluída. Quando um grupo de coisas dinâmico é atualizado, o status do grupo é alterado para `REBUILDING` e o grupo atualiza sua associação. Quando a alocação é concluída, o status muda para `ACTIVE`. Para verificar o status do seu grupo de coisas dinâmico, use o comando DescribeThingGroup.

Excluir um grupo de coisas dinâmicas

Use o comando DeleteDynamicThingGroup para excluir um grupo de coisas dinâmico:

```
$ aws iot delete-dynamic-thing-group --thing-group-name "RoomTooWarm"
```

O comando DeleteDynamicThingGroup não produz nenhuma saída.

Os comandos que mostram a quais grupos uma coisa pertence (por exemplo, ListGroupsForThing) podem continuar a mostrar o grupo enquanto os registros na nuvem estão sendo atualizados.

Limitações e conflitos

Grupos de coisas dinâmicas compartilham essas limitações com grupos de coisas estáticas:

- O número de atributos que um grupo de coisas pode ter é limitado.
- O número de grupos aos quais uma coisa pode pertencer é limitado.
- Grupos de coisas não podem ser renomeados.
- Nomes de coisas não podem conter caracteres internacionais, como û, é e ñ.

Ao usar grupos de coisas dinâmicos, tenha em mente o seguinte.

O serviço de indexação de frota deve ser habilitado

O serviço de indexação de frota deve ser habilitado e a alocação de indexação da frota deve ser concluída antes de você criar e usar grupos de coisas dinâmicos. Considere um atraso depois que você habilitar o serviço de indexação da frota. A alocação pode levar algum tempo para ser concluída. Quanto mais coisas que você tiver registrado, mais tempo levará o processo de alocação. Depois de ativar o serviço de indexação da frota para grupos de coisas dinâmicos, você não poderá desativá-lo até que todos os seus grupos de coisas dinâmicos sejam excluídos.

> **Note**
>
> Se você tiver permissões para consultar o índice de frota, você poderá acessar os dados das coisas em toda a frota.

O número de grupos de coisas dinâmicas é limitado

O número de grupos dinâmicos é limitado.

Comandos bem-sucedidos podem registrar erros

Ao criar ou atualizar um grupo de coisas dinâmicas, é possível que algumas coisas possam estar qualificadas para estar em um grupo dinâmico, mas ainda não tenham sido adicionadas a ele. O comando para criar ou atualizar um grupo de coisas dinâmicas, no entanto, ainda é bem-sucedido nesses casos ao registrar um erro e gerar uma métrica `AddThingToDynamicThingGroupsFailed` (p. 444).

UmaEntrada de log de errosNo log do CloudWatch é criado para cada coisa quando uma coisa qualificada não pode ser adicionada a um grupo de coisas dinâmicas ou uma coisa é removida de um grupo dinâmico

para ser adicionada a outro grupo. Quando uma coisa não pode ser adicionada a um grupo dinâmico, um`AddThingToDynamicThingGroupsFailed`métrica (p. 444)Também é criada. No entanto, uma única métrica pode representar várias entradas de log.

Quando uma coisa se qualifica para ser adicionada a um grupo de coisas dinâmicas, o seguinte é considerado:

- A coisa já está no máximo possível de grupos? (Consulte limites)
  - NÃO:A coisa é adicionada ao grupo dinâmico.
  - SIM: Essa coisa é um membro de algum grupo dinâmico de coisas dinâmicas?
    - NÃO: A coisa não pode ser adicionada ao grupo dinâmico, um erro é registrado e um`AddThingToDynamicThingGroupsFailed`métrica (p. 444)é gerado.
    - SIM: O grupo de coisas dinâmicas a ingressar é mais antigo do que qualquer grupo de coisas dinâmicas do qual a coisa já é membro?
      - NÃO: A coisa não pode ser adicionada ao grupo dinâmico, um erro é registrado e um`AddThingToDynamicThingGroupsFailed`métrica (p. 444)é gerado.
      - SIM: Remova a coisa do grupo dinâmico mais recente do qual é membro, registre um erro e adicione a coisa ao grupo de coisas dinâmicas. Isso gera um erro e uma métrica `AddThingToDynamicThingGroupsFailed` (p. 444) para o grupo dinâmico do qual a coisa foi removida.

Quando uma coisa em um grupo de coisas dinâmicas não satisfaz mais a consulta de pesquisa, ela é removida do grupo de coisas dinâmicas. Da mesma forma, quando uma coisa é atualizada para atender a consulta de pesquisa de um grupo de coisas dinâmicas, ela é adicionada ao grupo como descrito anteriormente. Essas adições e remoções são normais e não produzem entradas de log de erros.

Com `overrideDynamicGroups` habilitado, os grupos estáticos terão prioridade sobre os grupos dinâmicos

O número de grupos aos quais uma coisa pode pertencer é limitado. Quando você atualiza a associação de coisas usando os comandos AddThingToThingGroup ou UpdateThingGroupsForThing, adicionar o parâmetro `--overrideDynamicGroups` prioriza grupos de coisas estáticas sobre grupos de coisas dinâmicas.

Ao adicionar uma coisa a um grupo de coisas estáticas, o seguinte é considerado:

- A coisa já pertence ao número máximo de grupos?
  - NÃO: A coisa é adicionada ao grupo de coisas estáticas.
  - SIM: Está a coisa em algum grupo dinâmico?
    - NÃO: A coisa não pode ser adicionada ao grupo de coisas. O comando gera uma exceção.
    - SIM: Foi--overrideDynamicGroupshabilitado?
      - NÃO: A coisa não pode ser adicionada ao grupo de coisas. O comando gera uma exceção.
      - SIM: A coisa é removida do grupo dinâmico criado mais recentemente, um erro é registrado e um`AddThingToDynamicThingGroupsFailed`métrica (p. 444)É gerado para o grupo dinâmico do qual a coisa foi removida. Depois, a coisa é adicionada ao grupo de coisas estáticas.

Os grupos de coisas dinâmicos mais antigos têm prioridade sobre os mais recentes

O número de grupos aos quais uma coisa pode pertencer é limitado. Quando uma coisa se qualifica para ser adicionada a um grupo de coisas dinâmicas devido a uma operação de criação ou atualização, e a coisa já está no máximo possível de grupos, ela pode ser removida de outro grupo de coisas dinâmicas

para habilitar essa adição. Para obter mais informações sobre como isso ocorre, consulte Comandos bem-sucedidos podem registrar erros (p. 285) e Com `overrideDynamicGroups` habilitado, os grupos estáticos terão prioridade sobre os grupos dinâmicos (p. 286) para obter exemplos.

Quando uma coisa é removida de um grupo dinâmico, um erro é registrado e um evento é gerado.

Não é possível aplicar políticas a grupos de coisas dinâmicos

A tentativa de aplicar uma política a um grupo dinâmico gera uma exceção.

A associação do grupo de coisas dinâmico é eventualmente consistente

Somente o estado final de uma coisa é avaliada para o registro. Os estados intermediários podem ser ignorados se forem atualizados rapidamente. Evite associar uma regra ou um trabalho, a um grupo de coisas dinâmico cuja associação depende de um estado intermediário.

Marcar recursos do AWS IoT

Para ajudá-lo a gerenciar e organizar seus grupos de coisas, tipos de coisas, regras de tópicos, trabalhos, auditorias programadas e perfis de segurança, opcionalmente, você pode atribuir seus próprios metadados a cada um desses recursos na forma de tags. Esta seção descreve tags e mostra a você como criá-las.

Para ajudá-lo a gerenciar seus custos relacionados a coisas, você pode criar grupos de faturamento (p. 291) que contenham coisas. Depois, você pode atribuir tags que contenham metadados a cada um desses grupos de faturamento. Esta seção também discute os grupos de faturamento e os comandos disponíveis para criá-los e gerenciá-los.

Conceitos básicos de tags

Você pode usar tags para categorizar seus recursos da AWS IoT de diferentes formas (como por finalidade, por proprietário ou por ambiente). Isso é útil quando você tem muitos recursos do mesmo tipo — é possível identificar rapidamente um recurso baseado nas tags que você 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 seus tipos de coisas, o que ajuda a monitorar dispositivos por tipo. Recomendamos que você crie um conjunto de chave de tags que atenda às suas necessidades para cada tipo de recurso. Usar um conjunto consistente de chaves de tags facilita para você gerenciar seus recursos.

Você pode pesquisar e filtrar recursos de acordo com as tags que adicionar ou aplicar. Você também pode usar tags de grupos de faturamento para categorizar e rastrear seus custos. Também é possível usar tags para controlar o acesso aos recursos, conforme descrito em Utilização de tags com políticas do IAM (p. 289).

Para facilidade de uso, o Tag Editor (Editor de tags) noAWSO Management Console proporciona uma forma central unificada para criar e gerenciar suas tags. Para obter mais informações, consulteTrabalhar com o Tag EditoremTrabalhar com oAWSManagement Console.

Você também pode trabalhar com tags usando aAWS CLIO e aAWS IoTAPI. Você pode associar tags a grupos de coisas, tipos de coisas, regras de tópicos, trabalhos, perfis de segurança, políticas e grupos de faturamento ao criá-los usando a`Tags`campo nos seguintes comandos:

- CreateBillingGroup
- CriadoEstinação
- CreateDeviceProfile
- CreateDynamicThingGroup
- CreateJob
- CreateOTAUpdate
- CreatePolicy
- CreateScheduledAudit
- CreateSecurityProfile
- CreateServiceProfile
- CreateStream
- CreateThingGroup
- CreateThingType
- CreateTopicRule
- CreateWirelessGateway
- CreateWirelessDevice

Você pode adicionar, modificar ou excluir tags de recursos existentes que oferecem suporte a marcação, usando os seguintes comandos:

- TagResource
- ListTagsForResource
- UntagResource

Você pode editar chaves de tags e valores, e você pode remover as tags de um recurso a qualquer momento. Você pode 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.

Restrições e limitações 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 `aws:`Prefixo nos nomes ou valores das suas tags. 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 recurso.
- Se o seu esquema de tags é usado em vários serviços e recursos, lembre-se de que outros serviços talvez tenham restrições em caracteres permitidos. Os caracteres permitidos incluem letras, espaços e números representáveis em UTF-8, além dos seguintes caracteres especiais: + - = . _ : / @.

Utilização de tags com políticas do IAM

É possível aplicar permissões em nível de recurso baseadas em tags às políticas do IAM que você usa para as ações de API da AWS IoT. Isso oferece a você mais controle sobre quais recursos um usuário pode criar, modificar ou usar. Você pode usar o elemento `Condition` (também chamado bloco `Condition`) juntamente com os seguintes valores e chaves de contexto de condição em uma política do IAM para controlar o acesso do usuário (permissões) baseado em tags de um recurso:

- Use `aws: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 e as chaves de contexto de condição em uma política do IAM se aplicam somente às ações do AWS IoT em que um identificador de um recurso que pode ser marcado com tags é um parâmetro obrigatório. Por exemplo, o uso de DescribeEndpoint não é permitido ou negado com base em valores e chaves de contexto de condição porque nenhum recurso marcável (grupos de coisas, tipos de coisas, regras de tópicos, trabalhos ou perfil de segurança) está referenciado nesta solicitação. Para obter mais informações sobreAWS IoTrecursos que são etiquetáveis e

chaves de condição que eles suportam, leiaAções, recursos e chaves de condição para oAWS IoT.

Para obter mais informações sobre como usar tags, consulteComo controlar o acesso com tagsnoAWS Identity and Access ManagementGuia do usuário do. 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 para o`ThingGroup`ações. Um usuário do IAM restrito por essa política:

- Não é possível criar um grupo de coisas a tag “env=prod” (no exemplo, consulte a linha`"aws:RequestTag/env" : "prod"`).
- Não é possível modificar ou acessar um grupo de coisas que tenha uma tag “env=prod” (no exemplo, consulte a linha`"aws:ResourceTag/env" : "prod"`).

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": "iot:CreateThingGroup",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:RequestTag/env": "prod"
        }
      }
    },
    {
      "Effect": "Deny",
      "Action": [
        "iot:CreateThingGroup",
        "iot:DeleteThingGroup",
        "iot:DescribeThingGroup",
        "iot:UpdateThingGroup"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/env": "prod"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": [
        "iot:CreateThingGroup",
        "iot:DeleteThingGroup",
        "iot:DescribeThingGroup",
        "iot:UpdateThingGroup"
      ],
      "Resource": "*"
    }
  ]
}
```

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

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

Note

Se você permitir ou negar aos usuários o acesso a recursos com base em tags, considere 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.

Grupos de faturamento

A AWS IoT não permite que você aplique tags diretamente a coisas individuais, mas permite que você coloque as coisas em grupos de faturamento e aplique tags. Para a AWS IoT, a alocação de dados de custo e uso com base em tags são limitados a grupos de faturamento.

AWS IoT Corepara recursos LoRaWAN, como dispositivos sem fio e gateways, não podem ser adicionados aos grupos de faturamento. No entanto, eles podem ser associados aAWS IoTcoisas, que podem ser adicionadas aos grupos de faturamento.

Os seguintes comandos estão disponíveis:

- AddThingToBillingGroup adiciona uma coisa a um grupo de faturamento.
- CreateBillingGroup cria um grupo de faturamento.
- DeleteBillingGroup exclui o grupo de faturamento.
- DescribeBillingGroup retorna informações sobre um grupo de faturamento.
- ListBillingGroups lista os grupos de faturamento que você criou.
- ListThingsInBillingGroup lista as coisas que você adicionou a determinado grupo de faturamento.
- RemoveThingFromBillingGroup remove a coisa específica do grupo de faturamento.
- UpdateBillingGroup atualiza informações sobre o grupo de faturamento.
- CreateThing permite que você especifique um grupo de faturamento para a coisa ao criá-la.
- DescribeThing retorna a descrição de uma coisa, incluindo o grupo de faturamento ao qual ela pertence, se houver.

OAWS IoTA API sem fio fornece essas ações para associar dispositivos e gateways sem fio comAWS IoTcoisas.

- AssociateWirelessDevicewithThing
- AssociateWirelessGatewayWithThing

Visualizar alocação de custos e dados de uso

Você pode usar tags de grupos de faturamento para categorizar e rastrear seus custos. Quando você aplica tags a grupos de faturamento (e assim às coisas que eles incluem),AWSA gera um relatório de alocação de custos como um arquivo CSV (valor separado por vírgula) cujo uso e custos são agregados por 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 na alocação de custos, consulteUsar tags de alocação de custos donoAWSGuia do usuário de Billing and Cost Management do.

> **Note**
>
> Para associar com precisão os dados de uso e custo a essas coisas que você colocou em grupos de faturamento, cada dispositivo ou aplicativo deve:
>
> - Ser registrado como coisa na AWS IoT. Para obter mais informações, consulte Gerenciamento de dispositivos com o AWS IoT (p. 265).
> - Conecte-se ao agente de mensagem da AWS IoT por meio do MQTT usando apenas o nome da coisa, como o ID do cliente. Para obter mais informações, consulte the section called "Protocolos de comunicação de" (p. 83).
> - Autenticar-se usando um certificado de cliente associado à coisa.

As seguintes dimensões de preço estão disponíveis para grupos de faturamento (com base na atividade das coisas associadas ao grupo de faturamento):

- Conectividade (com base no nome da coisa usado como ID do cliente para se conectar).
- Sistema de mensagens (com base em mensagens de entrada e saída de uma coisa; MQTT apenas).
- Operações Shadow (com base na coisa cuja mensagem acionou uma atualização de shadow).
- Regras acionadas (com base na coisa cuja mensagem de entrada acionou a regra; não se aplica às regras acionadas por eventos de ciclo de vida de MQTT).
- Atualizações do índice de coisas (com base na coisa adicionada ao índice).
- Ações remotas (com base na coisa atualizada).
- Detectar (p. 993) relatórios (com base na coisa cuja atividade é reportada).

Os dados de custo e uso com base em tags (e reportados a um grupo de faturamento) não refletem as seguintes atividades:

- Operações de registro de dispositivos (incluindo atualizações de coisas, grupos de coisas e tipos de coisas). Para obter mais informações, consulte Gerenciamento de dispositivos com o AWS IoT (p. 265)).
- Atualizações de índice do grupo de coisas (ao adicionar um grupo de coisas).
- Consultas de pesquisa de índice.
- Provisionamento de dispositivos (p. 806).
- Relatórios Auditoria (p. 930).

Segurança em AWS IoT

A segurança da nuvem na AWS é a nossa maior prioridade. Como cliente da AWS, você se contará com um datacenter e uma arquitetura de rede criados para atender aos requisitos das organizações com as maiores exigências de segurança.

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

- Segurança da nuvem: a AWS é responsável pela proteção da infraestrutura que executa produtos da AWS na Nuvem AWS. A AWS também fornece serviços que podem ser usados com segurança. Auditores de terceiros testam e verificam regularmente a eficácia da nossa segurança como parte dos programas de conformidade da AWS. Para saber mais sobre os programas de conformidade que se aplicam ao AWS IoT, consulte Serviços da AWS no escopo pelo programa de conformidade.
- Segurança da nuvem: sua responsabilidade é determinada pelo serviço da AWS que você usa. Você também é responsável por outros fatores, incluindo a confidencialidade de seus dados, os requisitos da sua empresa e as leis e regulamentos aplicáveis.

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

Tópicos

- Segurança do AWS IoT (p. 293)
- Autenticação (p. 294)
- Autorização (p. 330)
- Proteção de dados no AWS IoT Core (p. 380)
- Identity and Access Management para o AWS IoT (p. 383)
- Registro em log e monitoramento (p. 420)
- Validação de conformidade doAWS IoTServiços (p. 421)
- Resiliência noAWSIoT Core (p. 422)
- Usar o AWS IoT Core com endpoints da VPC de interface (p. 422)
- Segurança da infraestrutura no AWS IoT (p. 425)
- Monitoramento de segurança de frotas de produção ou dispositivos comAWS IoTServiços (p. 425)
- Melhores práticas de segurança no AWS IoT Core (p. 425)
- Treinamento e certificação da AWS (p. 431)

Segurança do AWS IoT

Cada dispositivo ou cliente conectado deve ter uma credencial para interagir com o AWS IoT. Todo o tráfego de e paraAWS IoTO é enviado com segurança pelo Transport Layer Security (TLS).AWSOs mecanismos de segurança em nuvem protegem os dados à medida que eles seAWS IoTe outrasAWSServiços da .

- Você é responsável por gerenciar as credenciais do dispositivo (certificados X.509,AWScredenciais, identidades do Amazon Cognito, identidades federadas ou tokens de autenticação personalizados) e políticas emAWS IoT. Para obter mais informações, consulteGerenciamento de chaves no AWS IoT (p. 383) Você é responsável por atribuir identidades exclusivas a cada dispositivo e gerenciar as permissões para cada dispositivo ou grupo de dispositivos.
- Seus dispositivos se conectam aAWS IoTUsando certificados X.509 ou identidades do Amazon Cognito por meio de uma conexão TLS segura. Durante a pesquisa e o desenvolvimento, e no caso de alguns aplicativos que fazem chamadas de API ou usam WebSockets, também será possível autenticar usando os usuários e grupos do IAM ou tokens de autenticação personalizados. Para obter mais informações, consulteUsuários, grupos e funções do IAM (p. 317)
- Ao usar a autenticação do AWS IoT, o agente de mensagens é responsável por autenticar seus dispositivos, ingerir com segurança dados do dispositivo e conceder ou negar permissões de acesso especificadas para seus dispositivos usando políticas do AWS IoT.
- Ao usar a autenticação personalizada, um autorizador personalizado é responsável por autenticar seus dispositivos e conceder ou negar permissões de acesso especificadas para seus dispositivos usandoAWS IoTou políticas do IAM.
- OAWS IoTO mecanismo de regras encaminha dados do dispositivo para outros dispositivos ou a outrosAWSserviços de acordo com as regras que você define. Ele usa o AWS Identity and Access Management para transferir dados com segurança para o seu destino final. Para obter mais informações, consulteIdentity and Access Management para o AWS IoT (p. 383)

Autenticação

A autenticação é um mecanismo no qual você verifica a identidade de um cliente ou servidor. A autenticação do servidor é o processo em que os dispositivos ou outros clientes garantem que estão se comunicando com um endpoint real do AWS IoT. A autenticação do cliente é o processo em que os dispositivos ou outros clientes se autenticam com o AWS IoT.

Treinamento e certificação da AWS

Faça o curso a seguir para saber mais sobre autenticação emAWS IoT: Mergulhe profundamente noAWS IoTAutenticação e autorização.

Visão geral do certificado X.509

Os certificados X.509 são certificados digitais que usam a infraestrutura de chave pública X.509 padrão para associar uma chave pública a uma identidade contida em um certificado. Os certificados X.509 são

emitidos por uma entidade confiável chamada CA (autoridade de certificação). A CA mantém um ou mais certificados especiais chamados certificados CA que são usados para emitir certificados X.509. Somente a autoridade de certificação tem acesso aos certificados CA. As cadeias de certificados X.509 são usadas tanto para autenticação do servidor por clientes quanto para autenticação do cliente pelo servidor.

Autenticação do servidor

Quando seu dispositivo ou outro cliente tentar se conectar ao AWS IoT Core, o servidor do AWS IoT Core enviará um certificado X.509 que seu dispositivo usa para autenticar o servidor. A autenticação ocorre na camada TLS por meio da validação da cadeia de certificados X.509 (p. 298). Esse é o mesmo método usado pelo navegador ao acessar um URL HTTPS. Para usar certificados de sua própria autoridade de certificação, consulte Gerenciar os certificados CA (p. 301).

Quando seus dispositivos ou outros clientes estabelecem uma conexão TLS com um endpoint do AWS IoT Core, o AWS IoT Core apresenta uma cadeia de certificados que os dispositivos usam para verificar se estão se comunicando com o AWS IoT Core, e não com outro servidor se passando pelo AWS IoT Core. A cadeia apresentada depende de uma combinação do tipo de endpoint ao qual o dispositivo está se conectando e do conjunto de cifras (p. 381) que o cliente e o AWS IoT Core negociaram durante o handshake TLS.

Tipos de endpoint

O AWS IoT Core oferece suporte a dois tipos de endpoints de dados diferentes, `iot:Data` e `iot:Data-ATS`. Os endpoints `iot:Data` apresentam um certificado assinado pelo Certificado CA raiz G5 primário público classe 3 VeriSign. Os endpoints `iot:Data-ATS` apresentam um certificado de servidor assinado por uma CA do Amazon Trust Services.

Os certificados apresentados pelos endpoints ATS são assinados por Starfield. Algumas implementações de cliente TLS exigem a validação da raiz da confiança e exigem que os certificados da CA Starfield estejam instalados nos armazenamentos de confiança do cliente.

> **Warning**
>
> O uso de um método de fixação de certificado que adiciona hash a todo o certificado (incluindo o nome do emissor, etc.) não é recomendado porque isso fará com que a verificação do certificado falha, pois os certificados ATS que fornecemos são de assinatura cruzada Starfield e têm um nome de emissor diferente.

Use endpoints `iot:Data-ATS`, a menos que seu dispositivo exija certificados da CA Symantec ou Verisign. Os certificados Symantec e Verisign foram descontinuados e não têm mais suporte para a maioria dos navegadores da web.

Você pode usar o comando `describe-endpoint` para criar seu endpoint ATS.

```
aws iot describe-endpoint --endpoint-type iot:Data-ATS
```

O comando `describe-endpoint` retorna um endpoint no seguinte formato.

```
account-specific-prefix.iot.your-region.amazonaws.com
```

Na primeira vez que `describe-endpoint` for chamado, um endpoint será criado. Todas as chamadas subsequentes para `describe-endpoint` retornam o mesmo endpoint.

Para compatibilidade com versões anteriores, o AWS IoT Core ainda tem suporte de endpoints do Symantec. Para obter mais informações, consulte Como o AWS IoT Core está ajudando os clientes a lidar com a desconfiança futura das autoridades de certificação Symantec. Os dispositivos que operam com endpoints do ATS são totalmente inoperáveis com dispositivos que operam com endpoints do Symantec na mesma conta e não exigem registrar novamente.

> **Note**
>
> Para ver seu endpoint `iot:Data-ATS` no console do AWS IoT Core, escolha Settings (Configurações). O console exibe somente o endpoint `iot:Data-ATS`. Por padrão, o comando `describe-endpoint` exibe o endpoint `iot:Data` para compatibilidade com versões anteriores. Para ver o endpoint `iot:Data-ATS`, especifique o parâmetro `--endpointType`, como no exemplo anterior.

Como criar um `IotDataPlaneClient` com o AWS SDK para Java

Por padrão, o AWS SDK para Java versão 2 cria um `IotDataPlaneClient` usando um endpoint `iot:Data`. Para criar um cliente que use um endpoint `iot:Data-ATS`, faça o seguinte.

- Crie um endpoint `iot:Data-ATS` usando a API DescribeEndpoint .
- Especifique esse endpoint ao criar o `IotDataPlaneClient`.

O exemplo a seguir executa as duas operações.

```
public void setup() throws Exception {
        IotClient client =
 IotClient.builder().credentialsProvider(CREDENTIALS_PROVIDER_CHAIN).region(Region.US_EAST_1).build();
        String endpoint = client.describeEndpoint(r -> r.endpointType("iot:Data-
ATS")).endpointAddress();
        iot = IotDataPlaneClient.builder()
                                .credentialsProvider(CREDENTIALS_PROVIDER_CHAIN)
                                .endpointOverride(URI.create("https://" + endpoint))
                                .region(Region.US_EAST_1)
                                .build();
}
```

Certificados CA para autenticação do servidor

Dependendo do tipo de endpoint de dados que está usando e do conjunto de cifras negociado, os certificados de autenticação do servidor do AWS IoT Core são assinados por um dos seguintes certificados CA raiz:

Endpoints do VeriSign (legado)

- Chave RSA 2048 bit: Certificado CA raiz VeriSign Class 3 Public Principal G5

Endpoints do Amazon Trust Services (preferencial)

> **Note**
>
> Talvez seja necessário clicar com o botão direito do mouse nesses links e selecionar Salvar link como... para salvar esses certificados como arquivos.

- Chave RSA 2048 bit:Amazon Root CA 1.
- Chave RSA de 4096 bits: Amazon Root CA 2. Reservado para uso futuro.
- Chave ECC de 256 bits:Amazon Root CA 3.
- Chave ECC de 384 bits: Amazon Root CA 4. Reservado para uso futuro.

Esses certificados são todos assinados pelo Certificado da CA raiz Starfield. Todas as novas regiões do AWS IoT Core, a partir do lançamento de 9 de maio de 2018 do AWS IoT Core na região Ásia-Pacífico (Mumbai), fornecerão somente certificados ATS.

Diretrizes de autenticação do servidor

Há muitas variáveis que podem afetar a capacidade de um dispositivo de validar o certificado de autenticação do servidor do AWS IoT Core. Por exemplo, os dispositivos podem ter memória muito limitada para armazenar todos os certificados da CA raiz possíveis ou os dispositivos podem implementar um método não padrão de validação de certificado. Por estas razões, sugerimos seguir estas diretrizes:

- Recomendamos usar seu endpoint do ATS e instalar todos os compatíveisAmazon Root CACertificados.
- Se você não puder armazenar todos esses certificados em seu dispositivo e se os dispositivos não usarem a validação baseada em ECC, você poderá omitir oAmazon Root CA 3eAmazon Root CA 4Certificados ECC. Se seus dispositivos implementarem validação de certificado baseada em RSA, você poderá omitir oAmazon Root CA 1eAmazon Root CA 2Certificados RSA. Talvez seja necessário clicar com o botão direito do mouse nesses links e selecionar Salvar link como... para salvar esses certificados como arquivos.
- Se você estiver tendo problemas de validação de certificado do servidor ao se conectar ao endpoint do ATS, tente adicionar o certificado da CA raiz da Amazon relevante ao armazenamento confiável. Talvez seja necessário clicar com o botão direito do mouse nesses links e selecionar Salvar link como... para salvar esses certificados como arquivos.
  - Transassinatura cruzadaAmazon Root CA 1
  - Transassinatura cruzadaAmazon Root CA 2- Reservado para uso future.
  - Transassinatura cruzadaAmazon Root CA 3
  - Transassinatura cruzadaAmazon Root CA 4- Reservado para uso future.
- Se você estiver tendo problemas de validação de certificado do servidor, talvez seja necessário confiar explicitamente na CA raiz. Experimente adicionar oStarfield Root CA Certificatepara sua loja de confiança.
- Se você ainda tiver problemas depois de executar as etapas acima, entre em contato comAWSSuporte ao desenvolvedor.

> **Note**
>
> Os certificados CA têm uma data de expiração depois da qual não podem ser usados para validar um certificado de servidor. O certificados da CA podem precisar ser substituídos antes da data de expiração. Você deve verificar se é possível atualizar o certificado CA raiz em todos os seus dispositivos ou clientes para ajudar a garantir uma conectividade contínua e manter-se atualizado com as melhores práticas de segurança.

> **Note**
>
> Ao se conectar ao AWS IoT Core no código do dispositivo, passe o certificado para a API que você está usando para se conectar. A API usada varia de acordo com o SDK. Para obter mais informações, consulte os SDKs de dispositivos do AWS IoT Core (p. 1278).

Autenticação de cliente

O AWS IoT oferece suporte a três tipos de principais de identidade para autenticação do dispositivo ou do cliente:

- Certificados do cliente X.509 (p. 298)
- Usuários, grupos e funções do IAM (p. 317)
- Identidades do Amazon Cognito (p. 318)

Essas identidades podem ser usadas com dispositivos, aplicativos móveis, web ou de desktop. Elas podem até ser usadas por um usuário digitando comandos da interface de linha de comando (CLI) do AWS IoT. Normalmente,AWS IoTOs dispositivos usam certificados X.509, enquanto aplicativos móveis

usam identidades do Amazon Cognito. Aplicativos web e de desktop usam identidades do IAM ou federadas.AWS CLIcomandos usam o IAM. Para obter mais informações sobre as identidades do IAM, consulteIdentity and Access Management para o AWS IoT (p. 383).

Certificados do cliente X.509

Os certificados X.509 fornecem ao AWS IoT a capacidade de autenticar conexões de cliente e de dispositivo. Os certificados de cliente devem ser registrados no AWS IoT antes que um cliente possa se comunicar com o AWS IoT. Um certificado de cliente pode ser registrado em váriosConta da AWSs no mesmoRegião da AWSpara facilitar os dispositivos móveis entre seusConta da AWSs na mesma região. Consulte Usar certificados de cliente X.509 em váriosConta da AWSs com registro de várias contas (p. 299) para obter mais informações.

Recomendamos que cada dispositivo ou cliente receba um certificado exclusivo para habilitar ações de gerenciamento de clientes refinadas, incluindo a revogação de certificados. Os dispositivos e os clientes também devem oferecer suporte à rotação e à substituição de certificados para ajudar a garantir uma operação contínua conforme os certificados expiram.

Para obter informações sobre como usar certificados X.509 para oferecer suporte a alguns dispositivos, consulte Provisionamento de dispositivos (p. 806) para revisar as diferentes opções de gerenciamento e provisionamento de certificados compatíveis com o AWS IoT.

O AWS IoT oferece suporte a estes tipos de certificados X.509 de cliente:

- Certificados X.509 gerados pelo AWS IoT
- Certificados X.509 assinados por uma CA registrada no AWS IoT.
- Certificados X.509 assinados por uma CA que não está registrada no AWS IoT.

Esta seção descreve como gerenciar certificados X.509 no AWS IoT. É possível usar o console do AWS IoT ou a AWS CLI para executar estas operações de certificado:

- Criar certificados de cliente do AWS IoT (p. 300)
- Criar seus próprios certificados de cliente (p. 301)
- Registrar um certificado de cliente (p. 307)
- Ativar ou desativar um certificado de cliente (p. 311)
- Revogar um certificado de cliente (p. 314)

Para obter mais informações sobre os comandos da AWS CLI que executam essas operações, consulte Referência da CLI do AWS IoT.

Usar certificados do cliente X.509

Os certificados X.509 autenticam conexões de cliente e de dispositivo com o AWS IoT. Os certificados X.509 oferecem vários benefícios em relação a outros mecanismos de identificação e autenticação. Os certificados X.509 permitem que as chaves assimétricas sejam usadas com os dispositivos. Por exemplo, é possível gravar chaves privadas em um armazenamento seguro de um dispositivo para que o material criptográfico confidencial nunca saia do dispositivo. Os certificados X.509 fornecem opções mais fortes de autenticação de cliente em outros esquemas, como nome de usuário e senha ou tokens do portador, porque a chave privada nunca deixa o dispositivo.

O AWS IoT autentica os certificados usando o modo de autenticação do cliente do protocolo TLS. O suporte do TLS está disponível em muitas linguagens de programação e sistemas operacionais e é comumente usado para a criptografia de dados. Na autenticação de cliente TLS,AWS IoTSolicita um certificado X.509 do cliente e valida o status do certificado eConta da AWScontra um registro de certificados. Depois, ele desafia o cliente a provar a propriedade da chave privada que corresponde à

chave pública contida no certificado. O AWS IoT exige que os clientes enviem a Server Name Indication (SNI) extension para o protocolo TLS (Transport Layer Security). Para obter mais informações sobre como configurar a extensão SNI, consulte Segurança de transporte no AWS IoT (p. 381).

Os certificados X.509 podem ser verificados em relação a uma autoridade de certificação (CA) confiável. Você pode criar certificados de cliente que usam a CA raiz da Amazon e pode usar seus próprios certificados de cliente assinados por outra CA. Para obter mais informações sobre como usar seus próprios certificados X.509, consulte Criar seus próprios certificados de cliente (p. 301).

A data e a hora de quando os certificados assinados por um certificado CA expiram são definidas quando o certificado é criado. Os certificados X.509 gerados pelo AWS IoT expiram à meia-noite UTC em 31 de dezembro de 2049 (2049-12-31T23:59:59Z). Para obter mais informações sobre como usar o console do AWS IoT para criar certificados que usam a CA raiz da Amazon, consulte Criar certificados de cliente do AWS IoT (p. 300).

Usar certificados de cliente X.509 em váriosConta da AWSs com registro de várias contas

O registro em várias contas possibilita a movimentação de dispositivos entre oConta da AWSs na mesma Região ou em diferentes Regiões. Com isso, você pode registrar, testar e configurar um dispositivo em uma conta de pré-produção e, depois, registrar e usar o mesmo dispositivo e certificado de dispositivo em uma conta de produção. Você também pode registrar o certificado do cliente no dispositivo (os certificados de dispositivo) sem uma CA (p. 310) registrado com o AWS IoT.

> **Note**
>
> Certificados usados para registro em várias contas são suportados no`iot:Data-ATS`,`iot:Data`(legado)`iot:Jobs`, e`iot:CredentialProvider`Tipos de endpoint. Para obter mais informações sobreAWS IoTEndpoints de dispositivos, consulteAWS IoTdados do dispositivo e endpoints de serviço (p. 80).

> **Note**
>
> O registro em várias contas não oferece suporte just-in-time registro porque um certificado de verificação é necessário para registrar a CA, que é gerada apenas para uma conta específica.

Os dispositivos que usam o registro de várias contas devem enviar oExtensão de Server Name Indication (SNI)Para o protocolo TLS (Transport Layer Security) e forneça o endereço completo do endpoint no`host_name`campo, quando eles se conectam aAWS IoT.AWS IoTusa o endereço do endpoint em`host_name`para rotear a conexão para a corretaAWS IoTconta. Os dispositivos existentes que não enviarem um endereço de endpoint válido em `host_name` continuarão a funcionar, mas não poderão usar os recursos que exigem essas informações. Para obter mais informações sobre a extensão SNI e para saber como identificar o endereço de endpoint para o campo `host_name`, consulte Segurança de transporte no AWS IoT (p. 381).

Como usar o registro de várias contas

1. Você pode registrar os certificados do dispositivo sem uma CA. Consulte Registrar um certificado de cliente assinado por uma CA (CLI) não registrada (p. 310). O registro de uma CA é opcional. Não é necessário registrar a CA que assinou os certificados de dispositivo com oAWS IoT.
2. Você pode usar o correto`host_name`Na extensão SNI para TLS quando o dispositivo se conectar aoAWS IoT. Consulte Segurança de transporte no AWS IoT (p. 381).

Algoritmos de assinatura de certificados compatíveis com o AWS IoT

A AWS IoT oferece suporte aos seguintes algoritmos de assinatura de certificado:

- SHA256WITHRSA
- SHA384WITHRSA

- SHA512WITHRSA
- DSA_WITH_SHA256
- ECDSA-WITH-SHA256
- ECDSA-WITH-SHA384
- ECDSA-WITH-SHA512

Criar certificados de cliente do AWS IoT

O AWS IoT fornece certificados de cliente que são assinados pela autoridade de certificação (CA) raiz da Amazon.

Este tópico descreve como criar um certificado de cliente assinado pela autoridade de certificação raiz da Amazon e como fazer download dos arquivos do certificado. Depois de criar os arquivos do certificado de cliente, é necessário instalá-los no cliente.

> **Note**
>
> Cada certificado de cliente X.509 fornecido pelaAWS IoTcontém atributos de emissor e assunto definidos no momento da criação do certificado. Os atributos do certificado são imutáveis somente após a criação do certificado.

É possível usar o console do AWS IoT ou a AWS CLI para criar um certificado do AWS IoT assinado pela autoridade de certificação raiz da Amazon.

Criar um certificado do AWS IoT (console)

Como criar um certificado do AWS IoT usando o console do AWS IoT

1. Faça login noAWSO Management Console e abra aAWS IoTconsole.
2. No painel de navegação à esquerda, selecione Secure (Proteger), Certificates (Certificados) e Create (Criar).
3. Selecione One-click certificate creation (recommended) (Criação de certificado com um clique (recomendado)) – Create certificate (Criar certificado).
4. Na página Certificate created! (Certificado criado!), faça download dos arquivos do certificado de cliente da coisa, da chave pública e da chave privada para um local seguro. Esses certificados gerados peloAWS IoTestão disponíveis apenas para uso comAWS IoTServiços da .

   Se você também precisar do arquivo do certificado CA raiz da Amazon, essa página também tem o link da página de onde é possível fazer download dele.
5. Um certificado de cliente foi criado e registrado no AWS IoT. É necessário ativar o certificado antes de usá-lo em um cliente.

   SelecioneAtivarPara ativar o certificado do cliente agora. Se você não quiser ativar o certificado agora, Ativar um certificado de cliente (console) (p. 312) descreve como ativar o certificado posteriormente.
6. Se você quiser anexar uma política ao certificado, escolhaAnexar uma política.

   Se você não quiser anexar uma política agora, escolhaConcluídopara terminar. É possível anexar uma política posteriormente.

Depois de concluir o procedimento, instale os arquivos do certificado no cliente.

Criar um certificado do AWS IoT (CLI)

A AWS CLI fornece o comando create-keys-and-certificate para criar certificados de cliente assinados pela autoridade de certificação raiz da Amazon. Esse comando, no entanto, não faz download do arquivo de

certificado CA raiz da Amazon. É possível fazer download do arquivo de certificado CA raiz da Amazon em Certificados CA para autenticação do servidor (p. 296).

Esse comando cria arquivos de chave privada, chave pública e certificado X.509, além de registrar e ativar o certificado no AWS IoT.

```
aws iot create-keys-and-certificate \
    --set-as-active \
    --certificate-pem-outfile certificate_filename.pem \
    --public-key-outfile public_filename.key \
    --private-key-outfile private_filename.key
```

Se você não deseja ativar o certificado ao criá-lo e registrá-lo, esse comando cria os arquivos de chave privada, chave pública e certificado X.509 e registra o certificado, mas não o ativa. Ativar um certificado de cliente (CLI) (p. 312) descreve como ativar o certificado posteriormente.

```
aws iot create-keys-and-certificate \
    --no-set-as-active \
    --certificate-pem-outfile certificate_filename.pem \
    --public-key-outfile public_filename.key \
    --private-key-outfile private_filename.key
```

Instale os arquivos do certificado no cliente.

Criar seus próprios certificados de cliente

O AWS IoT oferece suporte a certificados de cliente assinados por outras autoridades de certificação (CA) raiz. É possível registrar certificados de cliente assinados por outra CA raiz; no entanto, se você deseja que o dispositivo ou o cliente registre o certificado de cliente quando ele se conectar pela primeira vez ao AWS IoT, a CA raiz deverá ser registrada no AWS IoT.

> **Note**
>
> Um certificado CA pode ser registrado por apenas uma conta em uma região.

Para obter mais informações sobre como usar certificados X.509 para oferecer suporte a alguns dispositivos, consulte Provisionamento de dispositivos (p. 806) para revisar as diferentes opções de gerenciamento e provisionamento de certificados compatíveis com o AWS IoT.

Tópicos

- Gerenciar os certificados CA (p. 301)
- Criar um certificado de cliente usando o certificado CA (p. 306)

Gerenciar os certificados CA

Esta seção descreve tarefas comuns para gerenciar seus próprios certificados de autoridade de certificação (CA).

Talvez seja necessário registrar a autoridade de certificação (CA) no AWS IoT caso você esteja usando certificados de cliente assinados por uma CA que o AWS IoT não reconhece.

Se você quiser que os clientes registrem automaticamente seus certificados de cliente no AWS IoT quando se conectam pela primeira vez, a CA que assinou os certificados de cliente deverá ser registrada no AWS IoT. Caso contrário, não é necessário registrar o certificado CA que assinou os certificados de cliente.

> **Note**
>
> Um certificado CA pode ser registrado por apenas uma conta em uma região.

- Criar um certificado CA (p. 302), se você precisar de um.

  Crie o certificado e os arquivos de chave necessários para a próxima etapa.
- Registrar o certificado CA (p. 302)

  Registrar o certificado CA comAWS IoT
- Desativar um certificado CA (p. 305)

Criar um certificado CA

Se você não tiver um certificado CA, pode usarOpenSSL v1.1iFerramentas para criar uma.

> **Note**
>
> Não é possível executar esse procedimento no console do AWS IoT.

Para criar um certificado CA usandoOpenSSL v1.1iferramentas

1. Gere um par de chaves.

```
openssl genrsa -out root_CA_key_filename.key 2048
```

2. Use a chave privada do par de chaves para gerar um certificado CA.

```
openssl req -x509 -new -nodes \
    -key root_CA_key_filename.key \
    -sha256 -days 1024 \
    -out root_CA_cert_filename.pem
```

Registrar o certificado CA

Esses procedimentos registram um certificado CAAWS IoT.

Registrar um certificado CA (console)

> **Note**
>
> Para registrar um certificado de CA no console, inicie no console emRegistrar certificado CA.

Registrar um certificado CA (CLI)

Certifique-se de que você tem o seguinte disponível no computador antes de continuar:

- O arquivo de certificado da CA raiz (referenciado abaixo como*root_CA_cert_filename.pem*)
- O arquivo de chave privada do certificado de CA raiz (referenciado abaixo como*root_CA_key_filename.key*)
- OpenSSL v1.1iou posterior

Como registrar um certificado CA usando a AWS CLI

1. Use get-registration-code para obter um código de registro do AWS IoT. Salve o `registrationCode` retornado para usar como o `Common Name` do certificado de verificação da chave privada.

```
aws iot get-registration-code
```

2. Gere um par de chaves para o certificado de verificação de chave privada:

```
openssl genrsa -out verification_cert_key_filename.key 2048
```

3. Crie uma solicitação de assinatura de certificado (CSR) para o certificado de verificação da chave privada. Defina o campo `Common Name` do certificado como o `registrationCode` retornado por get-registration-code.

```
openssl req -new \
    -key verification_cert_key_filename.key \
    -out verification_cert_csr_filename.csr
```

Você será solicitado a fornecer algumas informações, incluindo o `Common Name` para o certificado.

```
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:
    State or Province Name (full name) []:
    Locality Name (for example, city) []:
    Organization Name (for example, company) []:
    Organizational Unit Name (for example, section) []:
    Common Name (e.g. server FQDN or YOUR name) []:your_registration_code
    Email Address []:

    Please enter the following 'extra' attributes
    to be sent with your certificate request
    A challenge password []:
    An optional company name []:
```

4. Use uma CSR para criar o certificado de verificação de chave privada.

```
openssl x509 -req \
    -in verification_cert_csr_filename.csr \
    -CA root_CA_cert_filename.pem \
    -CAkey root_CA_key_filename.key \
    -CAcreateserial \
    -out verification_cert_filename.pem \
    -days 500 -sha256
```

5. Registre o certificado CA com a AWS IoT. Transmita o nome do arquivo do certificado CA e o nome do arquivo do certificado de verificação da chave privada para o comando register-ca-certificate:

```
aws iot register-ca-certificate \
    --ca-certificate file://root_CA_cert_filename.pem \
    --verification-cert file://verification_cert_filename.pem
```

Esse comando retornará o *`certificateId`*, se for bem-sucedido.

6. Neste momento, o certificado CA foi registrado no AWS IoT, mas não está ativo. O certificado CA deve ser ativado antes do registro de qualquer certificado de cliente assinado por ele.

Esta etapa ativa o certificado CA.

Use o comando da CLI update-certificate para ativar o certificado CA:

```
aws iot update-ca-certificate \
    --certificate-id certificateId \
    --new-status ACTIVE
```

Use o comando describe-ca-certificate para ver o status do certificado CA.

Criar um certificado de verificação da CA para registrar o certificado de CA no console

> **Note**
>
> Este procedimento é apenas para uso se você estiver registrando um certificado de CA a partir doAWS IoTconsole do .
> Se você não chegou a este procedimento a partir doAWS IoTconsole, inicie o processo de registro de certificado CA no console emRegistrar certificado CA.

Certifique-se de que você tem o seguinte disponível no mesmo computador antes de continuar:

- O arquivo de certificado da CA raiz (referenciado abaixo como*root_CA_cert_filename.pem*)
- O arquivo de chave privada do certificado de CA raiz (referenciado abaixo como*root_CA_key_filename.key*)
- OpenSSL v1.1.1iou posterior

Para usar a interface de linha de comando para criar um certificado de verificação de CA para registrar seu certificado de CA no console

1. Substituir*verification_cert_key_filename.key*Com o nome do arquivo de chave de certificado de veification que você deseja criar. Por exemplo,**verification_cert.key**E, em seguida, execute esse comando para gerar um key pair para o certificado de verificação de chave privada:

   ```
   openssl genrsa -out verification_cert_key_filename.key 2048
   ```

2. Substituir*verification_cert_key_filename.key*Com o nome do arquivo de chave que você criou na etapa 1.

   Substituir*verification_cert_csr_filename.csr*Com o nome do arquivo CSR (solicitação de assinatura de certificado) que você deseja criar. Por exemplo, **verification_cert.csr**.

   Execute este comando para criar o arquivo CSR.

   ```
   openssl req -new \
       -key verification_cert_key_filename.key \
       -out verification_cert_csr_filename.csr
   ```

   O comando solicita informações adicionais explicadas posteriormente.
3. NoAWS IoTConsole do, noCertificado de verificaçãocontainer, copie o código de registro.
4. A informação que oopensslO comando solicita que você faça o é mostrado no exemplo a seguir. Excepto pelo`Common Name`, você pode inserir seus próprios valores ou deixá-los em branco.

   No`Common Name`, cole o código de registro que você copiou na etapa anterior.

   ```
   You are about to be asked to enter information that will be incorporated
   into your certificate request.
   ```

```
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:
    State or Province Name (full name) []:
    Locality Name (for example, city) []:
    Organization Name (for example, company) []:
    Organizational Unit Name (for example, section) []:
    Common Name (e.g. server FQDN or YOUR name) []:your_registration_code
    Email Address []:

    Please enter the following 'extra' attributes
    to be sent with your certificate request
    A challenge password []:
    An optional company name []:
```

Depois de concluir, o comando cria o arquivo CSR.

5. Substituir*verification_cert_csr_filename.csr*com o*verification_cert_csr_filename.csr*Você usou na etapa anterior.

   Substituir*root_CA_cert_filename.pem*Com o nome do arquivo do certificado CA que você deseja registrar.

   Substituir*root_CA_key_filename.key*Com o nome do arquivo do arquivo de chave privada do certificado CA.

   Substituir*verification_cert_filename.pem*Com o nome de arquivo do certificado de verificação que você deseja criar. Por exemplo, **`verification_cert.pem`**.

```
openssl x509 -req \
    -in verification_cert_csr_filename.csr \
    -CA root_CA_cert_filename.pem \
    -CAkey root_CA_key_filename.key \
    -CAcreateserial \
    -out verification_cert_filename.pem \
    -days 500 -sha256
```

6. Depois que o comando openssl for concluído, você deverá ter esses arquivos prontos para uso quando retornar ao console.

   - Seu arquivo de certificado CA (*root_CA_cert_filename.pem*usado no comando anterior)
   - O certificado de verificação criado na etapa anterior (*verification_cert_filename.pem*usado no comando anterior)

Desativar um certificado CA

Quando um certificado de autoridade de certificação (CA) está habilitado para registro automático de certificado de cliente, o AWS IoT verifica o certificado CA usado para assinar o certificado de cliente a fim de verificar se a CA está `ACTIVE`. Se o certificado CA estiver `INACTIVE`, o AWS IoT não permitirá que o certificado de cliente seja registrado.

Ao definir o certificado CA como `INACTIVE`, você impede que os novos certificados de cliente emitidos pela CA sejam registrados automaticamente.

> **Note**
>
> Qualquer certificado de cliente registrado que foi assinado pelo certificado CA comprometido continuará a funcionar até que você revogue explicitamente cada um deles.

Desativar um certificado CA (console)

Como desativar um certificado CA usando o console do AWS IoT

1. Faça login noAWSConsole de Gerenciamento da e abra aAWS IoTconsole.
2. No painel de navegação à esquerda, selecione Secure (Proteger) e CAs.
3. Na lista de autoridades de certificação, localize a que pretende desativar e abra o menu de opções usando o ícone de reticências.
4. No menu de opções, selecione Deactivate (Desativar).

A autoridade de certificação deve aparecer como Inactive (Inativa) na lista.

> **Note**
>
> O console do AWS IoT não fornece uma maneira de listar os certificados que foram assinados pela CA que você desativou. Para obter uma opção da AWS CLI para listar esses certificados, consulte Desativar um certificado CA (CLI) (p. 306).

Desativar um certificado CA (CLI)

A AWS CLI fornece o comando update-ca-certificate para desativar um certificado CA.

```
aws iot update-ca-certificate \
    --certificate-id certificateId \
    --new-status INACTIVE
```

Use o comando list-certificates-by-ca para obter uma lista de todos os certificados de cliente registrados que foram assinados pela CA especificada. Para cada certificado de cliente assinado pelo certificado CA especificado, use o comando update-certificate para revogar o certificado de cliente a fim de impedir que ele seja usado.

Use o comando describe-ca-certificate para ver o status do certificado CA.

Criar um certificado de cliente usando o certificado CA

É possível usar sua própria autoridade de certificação (CA) para criar certificados de cliente. O certificado de cliente deve ser registrado no AWS IoT antes de ser usado. Para obter informações sobre as opções de registro para os certificados de cliente, consulte Registrar um certificado de cliente (p. 307).

Criar um certificado de cliente (CLI)

> **Note**
>
> Não é possível executar esse procedimento no console do AWS IoT.

Como criar um certificado de cliente usando a AWS CLI

1. Gere um par de chaves.

   ```
   openssl genrsa -out device_cert_key_filename.key 2048
   ```

2. Crie uma CSR para o certificado de cliente.

   ```
   openssl req -new \
       -key device_cert_key_filename.key \
       -out device_cert_csr_filename.csr
   ```

   Você será solicitado a fornecer algumas informações, conforme mostrado aqui.

```
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:
    State or Province Name (full name) []:
    Locality Name (for example, city) []:
    Organization Name (for example, company) []:
    Organizational Unit Name (for example, section) []:
    Common Name (e.g. server FQDN or YOUR name) []:
    Email Address []:

    Please enter the following 'extra' attributes
    to be sent with your certificate request
    A challenge password []:
    An optional company name []:
```

3. Crie um certificado de cliente usando a CSR.

```
openssl x509 -req \
    -in device_cert_csr_filename.csr \
    -CA root_CA_cert_filename.pem \
    -CAkey root_CA_key_filename.key \
    -CAcreateserial \
    -out device_cert_filename.pem \
    -days 500 -sha256
```

Neste momento, o certificado de cliente foi criado, mas ainda não foi registrado no AWS IoT. Para obter informações sobre como e quando registrar o certificado de cliente, consulte Registrar um certificado de cliente (p. 307).

Registrar um certificado de cliente

Os certificados de cliente devem ser registrados no AWS IoT para habilitar as comunicações entre o cliente e o AWS IoT. Você pode registrar cada certificado de cliente manualmente ou pode configurar os certificados de cliente para serem registrados automaticamente quando o cliente se conectar ao AWS IoT pela primeira vez.

Se você quiser que os clientes e os dispositivos registrem seus certificados de cliente quando se conectarem pela primeira vez, será necessário Registrar o certificado CA (p. 302) usado para assinar o certificado de cliente no AWS IoT nas regiões nas quais deseja usá-lo. A CA raiz da Amazon é registrada automaticamente no AWS IoT.

Os certificados de cliente podem ser compartilhados porConta da AWSs e regiões. Os procedimentos desses tópicos devem ser executados em cada conta e região em que você deseja usar o certificado de cliente. O registro de um certificado de cliente em uma conta ou região não é automaticamente reconhecido por outra.

> **Note**
>
> Os clientes que usam o protocolo TLS (Transport Layer Security) para se conectar ao AWS IoT devem oferecer suporte à Server Name Indication (SNI) extension para o TLS. Para obter mais informações, consulteSegurança de transporte no AWS IoT (p. 381)

Tópicos

- Registrar um certificado de cliente manualmente (p. 308)

- Registrar um certificado de cliente quando o cliente se conectar aoAWS IoT just-in-time Registro (JITR) (p. 310)

Registrar um certificado de cliente manualmente

É possível registrar um certificado de cliente manualmente usando o console do AWS IoT e a AWS CLI.

O procedimento de registro a ser usado depende se o certificado será compartilhado por compartilhado porConta da AWSs e regiões. O registro de um certificado de cliente em uma conta ou região não é automaticamente reconhecido por outra.

Os procedimentos deste tópico devem ser executados em cada conta e região em que você deseja usar o certificado de cliente. Os certificados de cliente podem ser compartilhados porConta da AWSs e Regiões, mas somente se o certificado de cliente for assinado por uma autoridade de certificação (CA) que NÃO esteja registrada noAWS IoT.

Registrar um certificado de cliente assinado por uma CA registrada (console)

> **Note**
>
> Antes de executar este procedimento, verifique se você tem o arquivo .pem do certificado de cliente e se o certificado de cliente foi assinado por uma CA registrada no AWS IoT (p. 302).

Como registrar um certificado existente no AWS IoT usando o console

1. Faça login noAWSConsole de Gerenciamento da e abra aAWS IoTconsole.
2. No painel de navegação à esquerda, selecione Secure (Proteger), Certificates (Certificados) e Create (Criar).
3. NoCriar um certificado, localize oUsar meu certificadoentrada e escolhaConceitos básicos.
4. Em Selecionar uma CA:

   - Se os certificados de cliente foram assinados por uma CA registrada noAWS IoT

     Escolha essa CA na lista e escolhaPróximo.
   - Se os certificados de cliente não foram assinados por uma CA registrada no AWS IoT

     Consulte Registrar um certificado de cliente assinado por uma CA não registrada (console) (p. 309).
   - Se os certificados de cliente forem assinados pela CA da Amazon

     Não selecione nenhuma CA, basta escolherPróximo.

   Se os certificados de cliente não foram assinados por uma CA registrada no AWS IoT, consulte Registrar um certificado de cliente assinado por uma CA não registrada (console) (p. 309).
5. NoRegistre certificados de dispositivo existentes, escolhaSelecionar certificadose selecione até 10 arquivos de certificado para registrar.
6. Depois de fechar a caixa de diálogo do arquivo, escolha se deseja ativar ou revogar os certificados de cliente ao registrá-los.

   Se você não ativar um certificado quando ele for registrado, Ativar um certificado de cliente (console) (p. 312) descreve como ativá-lo posteriormente.

   Se um certificado for revogado quando for registrado, ele não poderá ser ativado posteriormente.

   Depois de selecionar os arquivos de certificado a serem registrados e selecionar as ações a serem executadas após o registro, selecioneRegistrar certificados.

Os certificados de cliente registrados com êxito aparecem na lista de certificados.

Registrar um certificado de cliente assinado por uma CA não registrada (console)

> **Note**
>
> Antes de executar esse procedimento, verifique se você tem o arquivo .pem do certificado de cliente.

Como registrar um certificado existente no AWS IoT usando o console

1. Faça login noAWSConsole de Gerenciamento da e abra aAWS IoTconsole.
2. No painel de navegação à esquerda, selecione Secure (Proteger), Certificates (Certificados) e Create (Criar).
3. NoCriar um certificado, localize oUsar meu certificadoentrada e escolhaConceitos básicos.
4. NoSelecione uma CA, escolhaPróximo.
5. NoRegistre certificados de dispositivo existentes, escolhaSelecionar certificadose selecione até 10 arquivos de certificado para registrar.
6. Depois de fechar a caixa de diálogo do arquivo, escolha se deseja ativar ou revogar os certificados de cliente ao registrá-los.

   Se você não ativar um certificado quando ele for registrado, Ativar um certificado de cliente (console) (p. 312) descreve como ativá-lo posteriormente.

   Se um certificado for revogado quando for registrado, ele não poderá ser ativado posteriormente.

   Depois de selecionar os arquivos de certificado a serem registrados e selecionar as ações a serem executadas após o registro, selecioneRegistrar certificados.

Os certificados de cliente registrados com êxito aparecem na lista de certificados.

Registrar um certificado de cliente assinado por uma CA registrada (CLI)

> **Note**
>
> Antes de executar esse procedimento, verifique se você tem o arquivo .pem da autoridade de certificação (CA) e o arquivo .pem do certificado de cliente. O certificado de cliente deve ser assinado por uma autoridade de certificação (CA) registrada no AWS IoT (p. 302).

Use o comando register-certificate para registrar, mas não ativar, um certificado de cliente.

```
aws iot register-certificate \
    --certificate-pem file://device_cert_filename.pem \
    --ca-certificate-pem file://ca_cert_filename.pem
```

O certificado de cliente está registrado no AWS IoT, mas ainda não está ativo. Consulte Ativar um certificado de cliente (CLI) (p. 312) para obter informações sobre como ativá-lo posteriormente.

Também é possível ativar o certificado de cliente ao registrá-lo usando esse comando.

```
aws iot register-certificate \
    --set-as-active \
    --certificate-pem file://device_cert_filename.pem \
    --ca-certificate-pem file://ca_cert_filename.pem
```

Para obter mais informações sobre como ativar o certificado a fim de que ele possa ser usado para se conectar ao AWS IoT, consulte Ativar ou desativar um certificado de cliente (p. 311)

Registrar um certificado de cliente assinado por uma CA (CLI) não registrada

> **Note**
>
> Antes de executar esse procedimento, verifique se você tem o arquivo .pem do certificado.

Use o comando register-certificate-without-ca para registrar, mas não ativar, um certificado de cliente.

```
aws iot register-certificate-without-ca \
    --certificate-pem file://device_cert_filename.pem
```

O certificado de cliente está registrado no AWS IoT, mas ainda não está ativo. Consulte Ativar um certificado de cliente (CLI) (p. 312) para obter informações sobre como ativá-lo posteriormente.

Também é possível ativar o certificado de cliente ao registrá-lo usando esse comando.

```
aws iot register-certificate-without-ca \
    --status ACTIVE \
    --certificate-pem file://device_cert_filename.pem
```

Para obter mais informações sobre como ativar o certificado a fim de que ele possa ser usado para se conectar aoAWS IoTConsulte,Ativar ou desativar um certificado de cliente (p. 311).

Registrar um certificado de cliente quando o cliente se conectar aoAWS IoT just-in-time Registro (JITR)

É possível configurar um certificado CA para habilitar os certificados de cliente que ele assinou para se registrarem no AWS IoT automaticamente na primeira vez que o cliente se conectar ao AWS IoT.

Para registrar os certificados de cliente quando um cliente se conecta ao AWS IoT pela primeira vez, é necessário habilitar o certificado CA para registro automático e configurar a primeira conexão pelo cliente para fornecer os certificados necessários.

Configurar um certificado CA para oferecer suporte ao registro automático (console)

Como configurar um certificado CA para oferecer suporte ao registro automático do certificado de cliente usando o console do AWS IoT

1. Faça login noAWSConsole de Gerenciamento da e abra aAWS IoTconsole.
2. No painel de navegação à esquerda, selecione Secure (Proteger) e CAs.
3. Na lista de autoridades de certificação, localize aquela para a qual deseja ativar o registro automático e abra o menu de opções usando o ícone de reticências.
4. No menu de opções, selecione Enable auto-registration (Habilitar o registro automático).

> **Note**
>
> O status do registro automático não é mostrado na lista de autoridades de certificação. Para ver o status do registro automático de uma autoridade de certificação, é necessário abrir a página Details (Detalhes) da autoridade de certificação.

Configurar um certificado CA para oferecer suporte ao registro automático (CLI)

Se você já registrou o certificado CA no AWS IoT, use o comando update-ca-certificate para definir `autoRegistrationStatus` do certificado CA como `ENABLE`.

```
aws iot update-ca-certificate \
--certificate-id caCertificateId \
--new-auto-registration-status ENABLE
```

Se deseja habilitar `autoRegistrationStatus` quando registrar o certificado CA, use o comando register-ca-certificate.

```
aws iot register-ca-certificate \
--allow-auto-registration  \
--ca-certificate file://root_CA_cert_filename.pem \
--verification-cert file://verification_cert_filename.pem
```

Use o comando describe-ca-certificate para ver o status do certificado CA.

Configurar a primeira conexão feita por um cliente para registro automático

Quando um cliente tenta se conectar aAWS IoTPela primeira vez, deve apresentar um arquivo que contém o certificado de cliente assinado pelo certificado CA como parte do handshake TLS.

Quando o cliente se conecta aoAWS IoT, use o*client_certificate_filename*Arquivo como o arquivo de certificado.AWS IoTO reconhece o certificado CA como um certificado CA registrado, registra o certificado de cliente e define seu status como`PENDING_ACTIVATION`. Isso significa que o certificado de cliente foi registrado automaticamente e está aguardando ativação. O estado do certificado de cliente deve ser `ACTIVE` para que ele possa ser usado para se conectar ao AWS IoT.

> **Note**
>
> Você pode provisionar dispositivos usandoAWS IoT Core just-in-time recurso de registro (JITR) sem ter que enviar toda a cadeia de confiança na primeira conexão dos dispositivos paraAWS IoT Core. A apresentação do certificado da CA é opcional, mas o dispositivo é necessário para enviar oIndicação de nome de servidor (SNI)extensão quando eles se conectam.

Quando o AWS IoT registra automaticamente um certificado ou quando um cliente apresenta um certificado no status `PENDING_ACTIVATION`, o AWS IoT publica uma mensagem no seguinte tópico MQTT:

`$aws/events/certificates/registered/`*`caCertificateId`*

Em que *`caCertificateId`* é o ID do certificado CA que emitiu o certificado de cliente.

A mensagem publicada para este tópico tem a seguinte estrutura:

```
{
        "certificateId": "certificateId",
        "caCertificateId": "caCertificateId",
        "timestamp": timestamp,
        "certificateStatus": "PENDING_ACTIVATION",
        "awsAccountId": "awsAccountId",
        "certificateRegistrationTimestamp": "certificateRegistrationTimestamp"
}
```

Você pode criar uma regra que ouça esse tópico e execute algumas ações. Recomendamos que você crie uma regra do Lambda que verifique se o certificado de cliente não está em uma CRL (lista de revogação de certificados), que ative o certificado e crie e anexe uma política ao certificado. A política determina quais recursos o cliente pode acessar. Para obter mais informações sobre como criar uma regra do Lambda que ouve na`$aws/events/certificates/registered/`*`caCertificateID`*tópico e executa essas ações, consulteRegisto just-in-time de Certificados de Cliente emAWS IoT.

Se algum erro ou exceção ocorrer durante o registro automático dos certificados de cliente,AWS IoTenvia eventos ou mensagens para seus logs in CloudWatch Logs. Para obter mais informações sobre a configuração de logs para sua conta, consulteAmazônia CloudWatch documentação.

Ativar ou desativar um certificado de cliente

O AWS IoT verifica se um certificado de cliente está ativo quando autentica uma conexão.

É possível criar e registrar certificados de cliente sem ativá-los, para que eles não possam ser usados até que você queira usá-los. Também é possível desativar certificados de cliente ativos para desabilitá-los temporariamente. Por fim, é possível revogar certificados de cliente para impedir qualquer uso futuro deles.

Ativar um certificado de cliente (console)

Como ativar um certificado de cliente usando o console do AWS IoT

1. Faça login noAWSConsole de Gerenciamento da e abra aAWS IoTconsole.
2. No painel de navegação à esquerda, selecione Secure (Proteger) e Certificates (Certificados).
3. Na lista de certificados, localize o certificado que você deseja ativar e abra o menu de opções usando o ícone de reticências.
4. No menu de opções, selecione Activate (Ativar).

O certificado deve ser exibido como Active (Ativo) na lista de certificados.

Desativar um certificado de cliente (console)

Como desativar um certificado de cliente usando o console do AWS IoT

1. Faça login noAWSConsole de Gerenciamento da e abra aAWS IoTconsole.
2. No painel de navegação à esquerda, selecione Secure (Proteger) e Certificates (Certificados).
3. Na lista de certificados, localize o certificado que você deseja desativar e abra o menu de opções usando o ícone de reticências.
4. No menu de opções, selecione Deactivate (Desativar).

O certificado deve ser exibido como Inactive (Inativo) na lista de certificados.

Ativar um certificado de cliente (CLI)

A AWS CLI fornece o comando update-certificate para ativar um certificado.

```
aws iot update-certificate \
    --certificate-id certificateId \
    --new-status ACTIVE
```

Se o comando tiver sido bem-sucedido, o status do certificado será `ACTIVE`. Execute describe-certificate para ver o status do certificado.

```
aws iot describe-certificate \
    --certificate-id certificateId
```

Desativar um certificado de cliente (CLI)

A AWS CLI fornece o comando update-certificate para desativar um certificado.

```
aws iot update-certificate \
    --certificate-id certificateId \
    --new-status INACTIVE
```

Se o comando tiver sido bem-sucedido, o status do certificado será `INACTIVE`. Execute describe-certificate para ver o status do certificado.

```
aws iot describe-certificate \
    --certificate-id certificateId
```

Anexar uma coisa ou política a um certificado de cliente

Quando você criar e registrar um certificado separado de uma coisa do AWS IoT, ele não terá nenhuma política que autorize qualquer operação do AWS IoT, nem será associado a qualquer objeto de coisa do AWS IoT. Esta seção descreve como adicionar esses relacionamentos a um certificado registrado.

> **Important**
>
> Para concluir esses procedimentos, você já deve ter criado a coisa ou a política que deseja anexar ao certificado.

O certificado autentica um dispositivo comAWS IoTpara que ele possa se conectar. Anexar o certificado a um recurso coisa estabelece a relação entre o dispositivo (por meio do certificado) e o recurso coisa. Para autorizar o dispositivo a executarAWS IoTações, como permitir que o dispositivo se conecte e publique mensagens, uma política apropriada deve ser anexada ao certificado do dispositivo.

Anexar uma coisa a um certificado de cliente (console)

Será necessário o nome do objeto de coisa para concluir este procedimento.

Como anexar um objeto de coisa a um certificado registrado

1. Faça login noAWSConsole de Gerenciamento da e abra aAWS IoTconsole.
2. No painel de navegação à esquerda, selecione Secure (Proteger) e Certificates (Certificados).
3. Na lista de certificados, localize o certificado ao qual você deseja anexar uma política, abra o menu de opções do certificado escolhendo o ícone de reticências e escolha Anexar coisa.
4. No pop-up, localize o nome da coisa que você deseja anexar ao certificado, marque sua caixa de seleção e escolhaAnexar.

O objeto de coisa agora deve aparecer na lista de coisas na página de detalhes do certificado.

Anexar uma política a um certificado de cliente (console)

Será necessário o nome do objeto de política para concluir este procedimento.

Como anexar um objeto de política a um certificado registrado

1. Faça login noAWSConsole de Gerenciamento da e abra aAWS IoTconsole.
2. No painel de navegação à esquerda, selecione Secure (Proteger) e Certificates (Certificados).
3. Na lista de certificados, localize o certificado ao qual você deseja anexar uma política, abra o menu de opções do certificado escolhendo o ícone de reticências e escolha Anexar política.
4. No pop-up, localize o nome da política que você deseja anexar ao certificado, marque sua caixa de seleção e escolhaAnexar.

O objeto de política agora deverá ser exibido na lista de políticas na página de detalhes do certificado.

Anexar uma coisa a um certificado de cliente (CLI)

A AWS CLI fornece o comando attach-thing-principal para anexar um objeto de coisa a um certificado.

```
aws iot attach-thing-principal \
    --principal certificateArn \
    --thing-name thingName
```

Anexar uma política a um certificado de cliente (CLI)

A AWS CLI fornece o comando attach-policy para anexar um objeto de política a um certificado.

```
aws iot attach-policy \
    --target certificateArn \
    --policy-name policyName
```

Revogar um certificado de cliente

Se detectar atividade suspeita em um certificado de cliente registrado, você poderá revogá-lo para que ele não possa ser usado novamente.

Revogar um certificado de cliente (console)

Como revogar um certificado de cliente usando o console do AWS IoT

1. Faça login noAWSConsole de Gerenciamento da e abra aAWS IoTconsole.
2. No painel de navegação à esquerda, selecione Secure (Proteger) e Certificates (Certificados).
3. Na lista de certificados, localize o certificado que você deseja revogar e abra o menu de opções usando o ícone de reticências.
4. No menu de opções, selecione Revoke (Revogar).

Se o certificado tiver sido revogado com êxito, ele será exibido como Revoked (Revogado) na lista de certificados.

Revogar um certificado de cliente (CLI)

A AWS CLI fornece o comando update-certificate para revogar um certificado.

```
aws iot update-certificate \
    --certificate-id certificateId \
    --new-status REVOKED
```

Se o comando tiver sido bem-sucedido, o status do certificado será `REVOKED`. Execute describe-certificate para ver o status do certificado.

```
aws iot describe-certificate \
    --certificate-id certificateId
```

Transferir um certificado para outra conta

Certificados X.509 que pertencem a umConta da AWSpode ser transferido para outroConta da AWS.

Para transferir um certificado X.509 de umConta da AWSpara outro

1. the section called “Comece uma transferência de certificado” (p. 315)

   O certificado deve ser desativado e desanexado de todas as políticas e coisas antes de iniciar a transferência.
2. the section called “Aceitar ou rejeitar uma transferência de certificado” (p. 316)

   A conta receptora deve aceitar ou rejeitar explicitamente o certificado transferido. Depois que a conta receptora aceitar o certificado, o certificado deve ser ativado antes de usar.
3. the section called “Cancelar uma transferência de certificado” (p. 317)

   A conta de origem pode cancelar uma transferência, se o certificado não tiver sido aceito.

Comece uma transferência de certificado

Você pode começar a transferir um certificado para outroConta da AWSusando oAWS IoTconsoleou oAWS CLI.

Iniciar uma transferência de certificado (console)

Para concluir esse procedimento, você precisará do ID do certificado que você deseja transferir.

Faça este procedimento da conta com o certificado para transferir.

Para começar a transferir um certificado para outroConta da AWS

1. Faça login noAWSConsole de Gerenciamento da e abra aAWS IoTconsole.
2. No painel de navegação à esquerda, selecione Secure (Proteger) e Certificates (Certificados).

   Escolha o certificado com umAtivoouInativostatus que você deseja transferir e abra sua página de detalhes.
3. No certificadoDetalhesNa páginaAçõesmenu, se oDesativaropção está disponível, escolha a opçãoDesativarPara desativar o certificado.
4. No certificadoDetalhesNa página, no menu esquerdo, escolhaPolíticas.
5. No certificadoPolíticas, se houver alguma política anexada ao certificado, desconecte cada uma abrindo o menu de opções da política e escolhendoDesconectar.

   O certificado não deve possuir políticas anexadas antes de você continuar.
6. No certificadoPolíticasNa página, no menu esquerdo, escolhaCoisas.
7. No certificadoCoisas, se houver alguma coisa anexada ao certificado, desconecte cada uma abrindo o menu de opções da coisa e escolhendoDesconectar.

   O certificado não deve possuir coisas anexadas antes de continuar.
8. No certificadoCoisasNa página, no menu esquerdo, escolhaDetalhes.
9. No certificadoDetalhesNa páginaAçõesSelecione um menuIniciar transferênciapara abrir oIniciar transferênciaCaixa de diálogo.
10. NoIniciar transferênciacaixa de diálogo, insira oConta da AWSnúmero da conta para receber o certificado e uma mensagem curta opcional.
11. SelecioneIniciar transferênciapara transferir o certificado.

O console deve exibir uma mensagem que indique o êxito ou a falha da transferência. Se a transferência foi iniciada, o status do certificado será atualizado paraTransferido.

Iniciar uma transferência de certificado (CLI)

Para concluir esse procedimento, você precisará do*certificateId*O e a*certificateArn*do certificado que você deseja transferir.

Faça este procedimento da conta com o certificado para transferir.

Para começar a transferir um certificado para outroAWSconta

1. Usar aupdate-certificateComando para desativar o certificado.

   ```
   aws iot update-certificate --certificate-id certificateId --new-status INACTIVE
   ```
2. Desanexar todas as políticas.

   1. Usar alist-attached-policiesPara listar as políticas anexadas ao certificado.

```
aws iot list-attached-policies --target certificateArn
```

2. Para cada política anexada, use odetach-policycomando para desanexar a política.

```
aws iot detach-policy --target certificateArn --policy-name policy-name
```

3. Desanexar todas as coisas.

   1. Usar alist-principal-thingscomando para listar as coisas anexadas ao certificado.

```
aws iot list-principal-things --principal certificateArn
```

   2. Para cada coisa em anexo, use odetach-thing-principalcomando para desanexar a coisa.

```
aws iot detach-thing-principal --principal certificateArn --thing-name thing-name
```

4. Usar atransfer-certificatepara iniciar a transferência de certificados.

```
aws iot transfer-certificate --certificate-id certificateId --target-aws-
account account-id
```

Aceitar ou rejeitar uma transferência de certificado

Você pode aceitar ou rejeitar um certificado transferido para vocêConta da AWSde outroConta da AWSusando oAWS IoTconsoleou oAWS CLI.

Aceitar ou rejeitar uma transferência de certificado (console)

Para concluir esse procedimento, você precisará do ID do certificado que foi transferido para sua conta.

Faça este procedimento da conta que recebeu o certificado que foi transferido.

Para aceitar ou rejeitar um certificado que foi transferido para oConta da AWS

1. Faça login noAWSConsole de Gerenciamento da e abra aAWS IoTconsole.
2. No painel de navegação à esquerda, selecione Secure (Proteger) e Certificates (Certificados).

   Escolha o certificado com um status deTransferência pendenteQue você deseja aceitar ou rejeitar e abrir sua página de detalhes.
3. No certificadoDetalhesNa páginaAçõesmenu,

   - Para aceitar o certificado, escolhaAceite transferência.
   - Para não aceitar o certificado, escolhaTransferência de rejeição.

Aceitar ou rejeitar uma transferência de certificado (CLI)

Para concluir esse procedimento, você precisará do*certificateId*da transferência de certificado que você deseja aceitar ou rejeitar.

Faça este procedimento da conta que recebeu o certificado que foi transferido.

Para aceitar ou rejeitar um certificado que foi transferido para oConta da AWS

1. Usar aaccept-certificate-transfercomando para aceitar o certificado.

```
aws iot accept-certificate-transfer --certificate-id certificateId
```

2. Usar areject-certificate-transfercomando para rejeitar o certificado.

```
aws iot reject-certificate-transfer --certificate-id certificateId
```

Cancelar uma transferência de certificado

Você pode cancelar uma transferência de certificado antes que ela seja aceita usando oAWS IoTconsoleou oAWS CLI.

Cancelar uma transferência de certificado (console)

Para concluir esse procedimento, você precisará do ID da transferência de certificado que deseja cancelar.

Faça esse procedimento a partir da conta que iniciou a transferência de certificado.

Para cancelar uma transferência de certificado

1. Faça login noAWSConsole de Gerenciamento da e abra aAWS IoTconsole.
2. No painel de navegação à esquerda, selecione Secure (Proteger) e Certificates (Certificados).

   Escolha o certificado comTransferidostatus cuja transferência você deseja cancelar e abrir seu menu de opções.
3. No menu de opções do certificado, escolha oRevogar transferênciaopção para cancelar a transferência de certificado.

   > **Important**
   >
   > Tenha cuidado para não confundir oRevogar transferênciaopção com oRevogaropção. ORevogar transferênciaopção cancela a transferência do certificado, enquanto oRevogaropção torna o certificado irreversivelmente inutilizável porAWS IoT.

Cancelar uma transferência de certificado (CLI)

Para concluir esse procedimento, você precisará do*certificateId*da transferência de certificado que você deseja cancelar.

Faça esse procedimento a partir da conta que iniciou a transferência de certificado.

Usar acancel-certificate-transferpara cancelar a transferência de certificado.

```
aws iot cancel-certificate-transfer --certificate-id certificateId
```

Usuários, grupos e funções do IAM

Os usuários, grupos e funções do IAM são os mecanismos padrão para o gerenciamento de identidades e de autenticações noAWS. Você pode usá-los para se conectar aoAWS IoTInterfaces HTTP usando oAWSSDK eAWS CLI.

As funções do IAM também permitemAWS IoTPara acessar outrosAWSRecursos em sua conta em seu nome. Por exemplo, se você quiser que um dispositivo publique seu estado em uma tabela do DynamoDB, as funções do IAM permitirãoAWS IoTPara interagir com o Amazon DynamoDB. Para obter mais informações, consulte Funções do IAM.

Para conexões do corretor de mensagens por HTTP,AWS IoTO autentica usuários, grupos e funções do IAM usando o processo de assinatura Signature Version 4. Para obter mais informações, consulteSigningAWSSolicitações da API do.

Ao usar oAWSSignature versão 4 comAWS IoT, os clientes devem oferecer suporte ao seguinte na implementação do TLS:

- TLS 1.2, TLS 1.1, TLS 1.0.
- Validação de assinatura do certificado SHA-256 RSA.
- Um dos pacotes de criptografia na seção de suporte a pacotes de criptografia do TLS.

Para obter mais informações, consulte Identity and Access Management para o AWS IoT (p. 383).

Identidades do Amazon Cognito

O Amazon Cognito Identity permite criar privilégio limitado temporárioAWSCredenciais para uso em aplicativos móveis e web. Ao usar o Amazon Cognito Identity, crie grupos de identidades que criam identidades exclusivas para seus usuários e os autentica com provedores de identidade como Login with Amazon, Facebook e Google. Também é possível usar identidades do Amazon Cognito com suas próprias identidades autenticadas pelo desenvolvedor. Para obter mais informações, consulteAmazon Cognito Identity.

Para usar o Amazon Cognito Identity, você define um grupo de identidades do Amazon Cognito associado a uma função do IAM. A função do IAM está associada a uma política do IAM que concede às identidades do seu grupo de identidades a permissão para acessarAWSrecursos como chamadasAWSServiços da .

O Amazon Cognito Identity cria identidades não autenticadas e autenticadas. As identidades não autenticadas são usadas para usuários convidados em um aplicativo móvel ou web que desejam usar o aplicativo sem fazer login. Os usuários não autenticados recebem somente as permissões especificadas na política do IAM associada ao grupo de identidades.

Ao usar identidades autenticadas, além da política do IAM anexada ao grupo de identidades, é necessário anexar umAWS IoTpolítica para um Amazon Cognito Identity usando oAttachPolicyAPI e dê permissões a um usuário individual do seuAWS IoTaplicativo. Você pode usar oAWS IoTPolítica para atribuir permissões refinadas para clientes específicos e seus dispositivos.

Usuários autenticados e não autenticados são tipos de identidade diferentes. Se você não anexar umAWS IoTpolítica para o Amazon Cognito Identity, um usuário autenticado falha na autorização emAWS IoTe não tem acesso aoAWS IoTRecursos e ações. Para obter mais informações sobre como criar políticas para identidades do Amazon Cognito, consulteExemplos de política de publicação/inscrição (p. 350)eAutorização com identidades do Amazon Cognito (p. 373).

Autenticação personalizada

O AWS IoT Core permite definir autorizadores personalizados para que você possa gerenciar sua própria autenticação e autorização de cliente. Isso é útil quando você precisa usar mecanismos de autenticação diferentes daqueles queAWS IoT CoreO oferece suporte nativamente. (Para obter mais informações sobre os mecanismos suportados nativamente, consultethe section called “Autenticação de cliente” (p. 297)).

Por exemplo, se você estiver migrando dispositivos existentes no campo paraAWS IoT Coree esses dispositivos usam um token de portador personalizado ou nome de usuário e senha MQTT para autenticar, você pode migrá-los paraAWS IoT Coresem ter que fornecer novas identidades para eles. Você pode usar a autenticação personalizada com qualquer um dos protocolos de comunicação queAWS IoT CoreO oferece suporte a Para obter mais informações sobre os protocolos queAWS IoT Coresuporta, consultethe section called “Protocolos de comunicação de” (p. 83).

Tópicos

- Compreendendo o fluxo de trabalho de autenticação (p. 319)
- Criando e gerenciando autorizadores personalizados (p. 320)
- Conexão aoAWS IoT Coreusando autenticação personalizada (p. 326)
- Solução de problemas em seus autorizadores (p. 328)

Compreendendo o fluxo de trabalho de autenticação

A autenticação personalizada permite definir como autenticar e autorizar clientes usandorecursos do autorizador. Cada autorizador contém uma referência a uma função do Lambda gerenciada pelo cliente, uma chave pública opcional para validar credenciais do dispositivo e informações de configuração adicionais. O diagrama a seguir ilustra o fluxo de autorização para autenticação personalizada noAWS IoT Core.

AWS IoT CoreFluxo de trabalho personalizado de autorização e autenticação

A lista a seguir explica cada etapa no fluxo de trabalho de autenticação e autorização personalizados.

1. Um dispositivo se conecta a um clienteAWS IoT Coreendpoint de dados usando um dos compatíveisthe section called “Protocolos de comunicação de” (p. 83). O dispositivo passa credenciais nos campos de cabeçalho da solicitação ou nos parâmetros de consulta (para o HTTP Publish ou MQTT sobre WebSockets protocolos) ou no campo nome de usuário e senha da mensagem MQTT CONNECT (para os protocolos MQTT e MQTT sobre WebSockets).
2. AWS IoT Coreverifica uma das duas condições:
   - A solicitação de entrada especifica um autorizador.
   - OAWS IoT Coreponto final de dados que recebe a solicitação tem um autorizador padrão configurado para ela.

   SeAWS IoT CoreLocaliza um autorizador de uma dessas maneiras,AWS IoT Coreaciona a função do Lambda associada ao autorizador.
3. (Opcional) Se você ativou a assinatura de token,AWS IoT Corevalida a assinatura da solicitação usando a chave pública armazenada no autorizador antes de acionar a função do Lambda. Se a validação falhar,AWS IoT CorePara a solicitação sem invocar a função do Lambda.
4. A função Lambda recebe as credenciais e os metadados de conexão na solicitação e toma uma decisão de autenticação.
5. A função do Lambda retorna os resultados da decisão de autenticação e umAWS IoT Coredocumento de política que especifica quais ações são permitidas na conexão. A função Lambda também retorna informações que especificam com que frequênciaAWS IoT Corerevalida as credenciais na solicitação invocando a função do Lambda.
6. AWS IoT Coreavalia a atividade na conexão com a política que recebeu da função do Lambda.

Considerações sobre dimensionamento

Como uma função do Lambda lida com autenticação e autorização para seu autorizador, a função está sujeita aos limites de preço e serviço do Lambda, como taxa de execução simultânea. Para obter mais informações sobre preços do Lambda, consulteDefinição de preço do Lambda. Você pode gerenciar a carga em sua função do Lambda ajustando `orefreshAfterInSeconds`e`disconnectAfterInSeconds`parâmetros na resposta da função do Lambda. Para obter mais informações sobre o conteúdo da resposta da função do Lambda, consultethe section called “Definir sua função do Lambda” (p. 321).

> **Note**
>
> Se você deixar a assinatura ativada, poderá evitar o disparo excessivo do Lambda por clientes não reconhecidos. Considere isso antes de desabilitar o login em seu autorizador.

Criando e gerenciando autorizadores personalizados

AWS IoT Coreimplementa esquemas de autenticação e autorização personalizados usandorecursos do autorizador. Cada autorizador consiste nos seguintes componentes:

- Name (Nome): Uma única sequência definida pelo usuário que identifica o autorizador.
- ARN da função do Lambda.: O nome de recurso da Amazon (ARN) da função Lambda que implementa a lógica de autorização e de autenticação.
- Nome da chave do token: O nome da chave usado para extrair o token dos cabeçalhos HTTP, parâmetros de consulta ou nome de usuário MQTT CONNECT para executar a validação da assinatura. Esse valor é obrigatório se a assinatura estiver habilitada no autorizador.
- Sinalizador de assinatura desativado (opcional): Um valor booliano que especifica se é necessário desativar o requisito de assinatura nas credenciais. Isso é útil para cenários em que assinar as credenciais não faz sentido, como esquemas de autenticação que usam nome de usuário e senha do MQTT. O valor padrão é`false`Por isso, a assinatura está habilitada por padrão.

- Chave pública de assinatura de token: A chave pública queAWS IoT Coreusa para validar a assinatura do token. Seu comprimento mínimo é de 2.048 bits. Esse valor é obrigatório se a assinatura estiver habilitada no autorizador.

O Lambda cobra pelo número de vezes que sua função do Lambda é executada e pela quantidade de tempo que leva para que o código em sua função seja executado. Para obter mais informações sobre preços do Lambda, consulteDefinição de preço do Lambda. Para obter mais informações sobre como criar funções do Lambda, consulte aGuia do desenvolvedor do Lambda.

> **Note**
>
> Se você deixar a assinatura ativada, poderá evitar o disparo excessivo do Lambda por clientes não reconhecidos. Considere isso antes de desabilitar o login em seu autorizador.

Definir sua função do Lambda

QuandoAWS IoT Corechama seu autorizador, ele aciona o Lambda associado associado ao autorizador com um evento que contém o seguinte objeto JSON. O objeto JSON de exemplo contém todos os campos possíveis. Quaisquer campos que não sejam relevantes para a solicitação de conexão não estão incluídos.

```
{
    "token" :"aToken",
    "signatureVerified": Boolean, // Indicates whether the device gateway has validated the
 signature.
    "protocols": ["tls", "http", "mqtt"], // Indicates which protocols to expect for the
 request.
    "protocolData": {
        "tls" : {
            "serverName": "serverName" // The server name indication (SNI) host_name
 string.
        },
        "http": {
            "headers": {
                "#{name}": "#{value}"
            },
            "queryString": "?#{name}=#{value}"
        },
        "mqtt": {
            "username": "myUserName",
            "password": "myPassword", // A base64-encoded string.
            "clientId": "myClientId" // Included in the event only when the device sends
 the value.
        }
    },
    "connectionMetadata": {
        "id": UUID // The connection ID. You can use this for logging.
    },
}
```

A função Lambda deve usar essas informações para autenticar a conexão de entrada e decidir quais ações são permitidas na conexão. A função deve enviar uma resposta que contenha os seguintes valores.

- `isAuthenticated`: Um valor booliano que indica se a solicitação está autenticada.
- `principalId`: Uma string alfanumérica que atua como um identificador para o token enviado pela solicitação de autorização personalizada. O valor deve ser uma string alfanumérica com pelo menos um, e não mais que 128, caracteres e corresponder a esse padrão de expressão regular (regex):`([a-zA-Z0-9]){1,128}`.
- `policyDocuments`: Uma lista de formatados pelo JSONAWS IoT CorePara obter mais informações sobre como criarAWS IoT Core, consultethe section called “Políticas do AWS IoT Core” (p. 332). O

número máximo de documentos de política é de 10 documentos de política. Cada documento de política pode conter um máximo de 2.048 caracteres.

- `disconnectAfterInSeconds`: Um inteiro que especifica a duração máxima (em segundos) da conexão com aAWS IoT Coregateway. O valor mínimo é de 300 segundos e o valor máximo é de 86.400 segundos.
- `refreshAfterInSeconds`: Um inteiro que especifica o intervalo entre as atualizações de política. Quando esse intervalo passar,AWS IoT Coreinvoca a função do Lambda para permitir atualizações de políticas. O valor mínimo é de 300 segundos e o valor máximo é de 86.400 segundos.

O objeto JSON a seguir contém um exemplo de uma resposta que sua função do Lambda pode enviar.

```
{
"isAuthenticated":true, //A Boolean that determines whether client can connect.
"principalId": "xxxxxxxx",  //A string that identifies the connection in logs.
"disconnectAfterInSeconds": 86400,
"refreshAfterInSeconds": 300,
 "policyDocuments": [
      {
        "Version": "2012-10-17",
        "Statement": [
           {
              "Action": "iot:Publish",
              "Effect": "Allow",
              "Resource": "arn:aws:iot:us-east-1:<your_aws_account_id>:topic/
customauthtesting"
            }
         ]
       }
    ]
}
```

O`policyDocument`value deve conter um valor válidoAWS IoT Coredocumento da política. Para obter mais informações sobreAWS IoT Core, consultethe section called “Políticas do AWS IoT Core” (p. 332). Em conexões MQTT sobre TLS e MQTT sobre WebSockets,AWS IoT Corearmazena em cache essa política para o intervalo especificado no valor do`refreshAfterInSeconds`campo. No caso de conexões HTTP, a função do Lambda é chamada para cada solicitação de autorização, a menos que seu dispositivo esteja usando conexões persistentes HTTP (também chamadas de manutenção de atividade HTTP ou reutilização de conexão HTTP), você pode optar por ativar o armazenamento em cache ao configurar o autorizador. Durante esse intervalo,AWS IoT Coreautoriza ações em uma conexão estabelecida contra essa política armazenada em cache sem acionar sua função do Lambda novamente. Se ocorrerem falhas durante a autenticação personalizada,AWS IoT Coreencerra a conexão.AWS IoT Coretambém encerra a conexão se ela estiver aberta por mais tempo do que o valor especificado na`disconnectAfterInSeconds`parâmetro .

Os seguintes exemplos de JavaScript contém um exemplo de função do Lambda Node.js que procura uma senha na mensagem MQTT Connect com um valor de`teste` retorna uma política que concede permissão para se conectar àAWS IoT Corecom um cliente chamado`myClientName`e publique em um tópico que contém o mesmo nome de cliente. Se não encontrar a senha esperada, ela retornará uma política que nega essas duas ações.

```
// A simple Lambda function for an authorizer. It demonstrates
// how to parse an MQTT password and generate a response.

exports.handler = function(event, context, callback) {
    var uname = event.protocolData.mqtt.username;
    var pwd = event.protocolData.mqtt.password;
    var buff = new Buffer(pwd, 'base64');
```

```
    var passwd = buff.toString('ascii');
    switch (passwd) {
        case 'test':
            callback(null, generateAuthResponse(passwd, 'Allow'));
        default:
            callback(null, generateAuthResponse(passwd, 'Deny'));
    }
};

// Helper function to generate the authorization response.
var generateAuthResponse = function(token, effect) {
    var authResponse = {};
    authResponse.isAuthenticated = true;
    authResponse.principalId = 'TEST123';

    var policyDocument = {};
    policyDocument.Version = '2012-10-17';
    policyDocument.Statement = [];
    var publishStatement = {};
    var connectStatement = {};
    connectStatement.Action = ["iot:Connect"];
    connectStatement.Effect = effect;
    connectStatement.Resource = ["arn:aws:iot:us-east-1:123456789012:client/myClientName"];
    publishStatement.Action = ["iot:Publish"];
    publishStatement.Effect = effect;
    publishStatement.Resource = ["arn:aws:iot:us-east-1:123456789012:topic/telemetry/
myClientName"];
    policyDocument.Statement[0] = connectStatement;
    policyDocument.Statement[1] = publishStatement;
    authResponse.policyDocuments = [policyDocument];
    authResponse.disconnectAfterInSeconds = 3600;
    authResponse.refreshAfterInSeconds = 300;

    return authResponse;
}
```

A função Lambda anterior retorna o seguinte JSON quando recebe a senha esperada do`test`na mensagem MQTT Connect. Os valores do`password`e`principalId`das propriedades serão os valores da mensagem MQTT Connect.

```
{
  "password": "password",
  "isAuthenticated": true,
  "principalId": "principalId",
  "policyDocuments": [
    {
      "Version": "2012-10-17",
      "Statement": [
        {
          "Action": "iot:Connect",
          "Effect": "Allow",
          "Resource": "*"
        },
        {
          "Action": "iot:Publish",
          "Effect": "Allow",
          "Resource": "arn:aws:region:accountId:topic/telemetry/${iot:ClientId}"
        },
        {
          "Action": "iot:Subscribe",
          "Effect": "Allow",
          "Resource": "arn:aws:iot:region:accountId:topicfilter/telemetry/${iot:ClientId}"
        },
```

```
        {
          "Action": "iot:Receive",
          "Effect": "Allow",
          "Resource": "arn:aws:iot:region:accountId:topic/telemetry/${iot:ClientId}"
        }
      ]
    }
  ],
  "disconnectAfterInSeconds": 3600,
  "refreshAfterInSeconds": 300
}
```

Criar um autorizador

Você pode criar um autorizador usando aCreateAuthorizer API. O exemplo a seguir descreve o comando.

```
aws iot create-authorizer
--authorizer-name MyAuthorizer
--authorizer-function-arn arn:aws:lambda:us-
west-2:<account_id>:function:MyAuthorizerFunction  //The ARN of the Lambda function.
[--token-key-name MyAuthorizerToken //The key used to extract the token from headers.
[--token-signing-public-keys FirstKey=
 "-----BEGIN PUBLIC KEY-----
  [...insert your public key here...]
  -----END PUBLIC KEY-----"
[--status ACTIVE]
[--tags <value>]
[--signing-disabled | --no-signing-disabled]
```

Você pode usar `osigning-disabled`parâmetro para desativar a validação de assinatura para cada invocação de seu autorizador. Recomendamos fortemente que você não desabilite a assinatura a menos que seja necessário. A validação de assinatura protege você contra invocações excessivas de sua função do Lambda de dispositivos desconhecidos. Não é possível atualizar `osigning-disabled`status de um autorizador depois de criá-lo. Para alterar esse comportamento, você deve criar outro autorizador personalizado com um valor diferente para `osigning-disabled`parâmetro .

Valores do`tokenKeyName`e`tokenSigningPublicKeys`os parâmetros são opcionais se você tiver desativado a assinatura. Eles são valores obrigatórios se a assinatura estiver ativada.

Depois de criar sua função do Lambda e o autorizador personalizado, você deve conceder explicitamente oAWS IoT Corepermissão de serviço para invocar a função em seu nome. É possível fazer isso com o comando a seguir.

```
aws lambda add-permission --function-name <lambda_function_name> --principal
iot.amazonaws.com --source-arn <authorizer_arn> --statement-id Id-123 --action
"lambda:InvokeFunction"
```

Testar autorizadores

Você pode usar oTestInvokeAuthorizerAPI para testar os valores de invocação e retornar do autorizador. Essa API permite que você especifique metadados de protocolo e teste a validação da assinatura em seu autorizador.

As guias a seguir mostram como usar oAWS CLIpara testar seu autorizador.

Unix-like

```
aws iot test-invoke-authorizer --authorizer-name NAME_OF_AUTHORIZER \
--token TOKEN_VALUE --token-signature TOKEN_SIGNATURE
```

Windows CMD

```
aws iot test-invoke-authorizer --authorizer-name NAME_OF_AUTHORIZER ^
--token TOKEN_VALUE --token-signature TOKEN_SIGNATURE
```

Windows PowerShell

```
aws iot test-invoke-authorizer --authorizer-name NAME_OF_AUTHORIZER `
--token TOKEN_VALUE --token-signature TOKEN_SIGNATURE
```

O valor do`token-signature`parâmetro é o token assinado. Para saber como obter esse valor, consultethe section called “Assinar o token” (p. 327).

Se o autorizador usar um nome de usuário e uma senha do, você poderá transmitir essas informações usando o`--mqtt-context`parâmetro . As guias a seguir mostram como usar o`TestInvokeAuthorizer`API para enviar um objeto JSON que contém um nome de usuário, senha e nome de cliente para o autorizador personalizado.

Unix-like

```
aws iot test-invoke-authorizer --authorizer-name NAME_OF_AUTHORIZER  \
--mqtt-context '{"username": "USER_NAME", "password": "dGVzdA==",
 "clientId":"CLIENT_NAME"}'
```

Windows CMD

```
aws iot test-invoke-authorizer --authorizer-name NAME_OF_AUTHORIZER  ^
--mqtt-context '{"username": "USER_NAME", "password": "dGVzdA==",
 "clientId":"CLIENT_NAME"}'
```

Windows PowerShell

```
aws iot test-invoke-authorizer --authorizer-name NAME_OF_AUTHORIZER  `
--mqtt-context '{"username": "USER_NAME", "password": "dGVzdA==",
 "clientId":"CLIENT_NAME"}'
```

A senha deve ser codificada em base64. O exemplo a seguir mostra como codificar uma senha em um ambiente semelhante ao UNIX.

```
echo -n PASSWORD | base64
```

Gerenciar autorizadores personalizados

Você pode gerenciar seus autorizadores usando as APIs a seguir.

- ListAuthorizers: Mostrar todos os autorizadores em sua conta.
- DescribeAuthorizer: Exibe propriedades do autorizador especificado. Esses valores incluem data de criação, data da última modificação e outros atributos.
- SetDefaultAuthorizer: Especifica o autorizador padrão do seuAWS IoT Coreendpoints de dados.AWS IoT Coreusa este autorizador se um dispositivo não passarAWS IoT Corecredenciais e não especifica um autorizador. Para obter mais informações sobre como usarAWS IoT Corecredenciais, consultethe section called “Autenticação de cliente” (p. 297).

- UpdateAuthorizer: Altera o status, o nome da chave do token ou as chaves públicas do autorizador especificado.
- DeleteAuthorizer: Exclui o autorizador especificado.

> **Note**
>
> Você não pode atualizar o requisito de assinatura de um autorizador. Isso significa que você não pode desativar o login em um autorizador existente que o exija. Você também não pode exigir o login em um autorizador existente que não o exija.

Conexão aoAWS IoT Coreusando autenticação personalizada

Os dispositivos podem se conectar aAWS IoT Coreusando autenticação personalizada com qualquer protocolo queAWS IoT Coresuporte para mensagens de dispositivos. Para obter mais informações sobre protocolos de comunicação compatíveis, consultethe section called “Protocolos de comunicação de” (p. 83). Os dados de conexão que você passa para a função do autorizador Lambda dependem do protocolo que você usa. Para obter mais informações sobre como criar sua função do autorizador do Lambda, consultethe section called “Definir sua função do Lambda” (p. 321). As seções a seguir explicam como se conectar para autenticar usando cada protocolo com suporte.

HTTP

Dispositivos que enviam dados paraAWS IoT Coreusando oAPI de publicação HTTPpode passar credenciais por meio de cabeçalhos de solicitação ou parâmetros de consulta em suas solicitações HTTP POST. Os dispositivos podem especificar um autorizador a ser invocado usando o`x-amz-customauthorizer-name`parâmetro de cabeçalho ou consulta. Se você tiver a assinatura de token ativada em seu autorizador, você deve passar o*`token-key-name`*`x-amz-customauthorizer-signature`em cabeçalhos de solicitação ou parâmetros de consulta. Observe que o*`token-signature`*value deve ser codificado por URL ao usar JavaScript De dentro do navegador.

As solicitações de exemplo a seguir mostram como você passa esses parâmetros nos cabeçalhos de solicitação e nos parâmetros de consulta.

```
//Passing credentials via headers
POST /topics/topic?qos=qos HTTP/1.1
Host: your-endpoint
x-amz-customauthorizer-signature: token-signature
token-key-name: token-value
x-amz-customauthorizer-name: authorizer-name

//Passing credentials via query parameters
POST /topics/topic?qos=qos&x-amz-customauthorizer-signature=token-signature&token-key-name=token-value HTTP/1.1
```

MQTT

Dispositivos que se conectam aoAWS IoT Coreusando uma conexão MQTT pode passar credenciais através do`username`e`password`campos de mensagens MQTT. O`username`O valor também pode, opcionalmente, conter uma string de consulta que transmite valores adicionais (incluindo um token, assinatura e nome do autorizador) para o seu autorizador. Você pode usar essa string de consulta se quiser usar um esquema de autenticação baseado em token em vez de`username`e`password`Valores.

> **Note**
>
> Os dados no campo de senha são codificados por base64 porAWS IoT Core. Sua função do Lambda deve decodificá-la.

O exemplo a seguir contém um`username`String que contém parâmetros extras que especificam um token e uma assinatura.

```
username?x-amz-customauthorizer-name=authorizer-name&x-amz-customauthorizer-
signature=token-signature&token-key-name=token-value
```

Para invocar um autorizador, dispositivos que se conectam aAWS IoT Coreusando MQTT e a autenticação personalizada deve se conectar na porta 443. Eles também devem passar a extensão do TLS Application Layer Protocol Negotiation (ALPN) com um valor de`mqtt`e a extensão SNI (Server Name Indication, indicação de nome de servidor) com o nome de host doAWS IoT Coreendpoint de dados. Para obter mais informações sobre esses valores, consultethe section called “Protocolos de comunicação de” (p. 83). O V2AWS IoTSDKs de dispositivos, SDKs móveis eAWS IoTCliente de dispositivo (p. 1278)pode configurar essas duas extensões.

MQTT pelo WebSockets

Dispositivos que se conectam aoAWS IoT CoreUsando MQTT por WebSockets Pode transmitir credenciais de uma das duas maneiras a seguir.

- Através de cabeçalhos de solicitação ou parâmetros de consulta na solicitação HTTP UPGRADE para estabelecer o WebSockets conexão.
- Através do`username`e`password`campos na mensagem MQTT CONNECT.

Se você passar credenciais por meio da mensagem de conexão MQTT, as extensões ALPN e SNI TLS serão necessárias. Para obter mais informações sobre essas extensões, consultethe section called “MQTT” (p. 326). O exemplo a seguir demonstra como transmitir credenciais pela solicitação de atualização HTTP.

```
GET /mqtt HTTP/1.1
Host: your-endpoint
Upgrade: WebSocket
Connection: Upgrade
x-amz-customauthorizer-signature: token-signature
token-key-name: token-value
sec-WebSocket-Key: any random base64 value
sec-websocket-protocol: mqtt
sec-WebSocket-Version: websocket version
```

Assinar o token

Você deve assinar o token com a chave privada do key pair pública/privada usado na`create-authorizer`Chame. Os exemplos a seguir mostram como criar a assinatura do token usando um comando do tipo UNIX e JavaScript. Eles usam o algoritmo hash SHA-256 para codificar a assinatura.

Command line

```
echo -n TOKEN_VALUE | openssl dgst -sha256 -sign PEM encoded RSA private key | openssl
 base64
```

JavaScript

```
const crypto = require('crypto')

const key = "PEM encoded RSA private key"

const k = crypto.createPrivateKey(key)
let sign = crypto.createSign('SHA256')
sign.write(t)
sign.end()
```

```
const s = sign.sign(k, 'base64')
```

Solução de problemas em seus autorizadores

Este tópico aborda problemas comuns que podem causar problemas em fluxos de trabalho de autenticação personalizados e etapas para resolvê-los. Para solucionar problemas de forma mais eficaz, ative CloudWatch Logs do para oAWS IoT Coree defina o nível de log comoDEPURAR. Você pode ativar CloudWatch logs noAWS IoT CoreConsole do (https://console.aws.amazon.com/iot/). Para obter mais informações sobre como habilitar e configurar logs paraAWS IoT CoreConsulte,the section called “Configurar registro em log da AWS IoT” (p. 433).

> **Note**
>
> Se você deixar o nível de log emDEPURARpor longos períodos de tempo, CloudWatch pode armazenar grandes quantidades de dados de registro em log. Isso pode aumentar suas cobranças do CloudWatch. Considere usar o registro baseado em recursos para aumentar a verbosidade apenas para dispositivos em um determinado grupo de coisas. Para obter mais informações sobre log baseado em recursos, consultethe section called “Configurar registro em log da AWS IoT” (p. 433). Além disso, quando você terminar a solução de problemas, reduza o nível de log para um nível menos detalhado.

Antes de começar a solucionar problemas, revisethe section called “Compreendendo o fluxo de trabalho de autenticação” (p. 319)Para obter uma exibição de alto nível do processo de autenticação personalizada. Isso ajuda você a entender onde procurar a origem de um problema.

Este tópico discute as duas áreas a seguir para você investigar.

- Problemas relacionados à função Lambda do seu autorizador.
- Problemas relacionados ao seu dispositivo.

Verifique se há problemas na função Lambda do autorizador

Execute as etapas a seguir para garantir que as tentativas de conexão de seus dispositivos estejam invocando sua função do Lambda.

1. Verifique qual função do Lambda está associada ao autorizador.

   Você pode fazer isso chamando oDescribeAuthorizerAPI ou clicando no autorizador desejado naSecureSeção do doAWS IoT Coreconsole do .
2. Verifique as métricas de invocação para a função do Lambda. Execute as etapas a seguir para fazer isso.

   a. Abrir oAWS LambdaConsole do (https://console.aws.amazon.com/lambda/) e selecione a função que está associada ao autorizador.

   b. Selecione oMonitormétricas de guia e visualização para o período de tempo relevante para o seu problema.
3. Se você não vir invocações, verifique seAWS IoT CoreO tem permissão para chamar sua função do Lambda. Se você vir chamadas, vá para a próxima etapa. Execute as etapas a seguir para verificar se sua função do Lambda tem as permissões necessárias.

   a. Selecione oPermissõesguia para sua função noAWS Lambdaconsole do .

   b. Encontre oPolítica baseada em recursona parte inferior da página. Se sua função do Lambda tiver as permissões necessárias, a política se parecerá com o exemplo a seguir.

```
{
  "Version": "2012-10-17",
  "Id": "default",
  "Statement": [
    {
      "Sid": "Id123",
      "Effect": "Allow",
      "Principal": {
        "Service": "iot.amazonaws.com"
      },
      "Action": "lambda:InvokeFunction",
      "Resource": "arn:aws:lambda:us-east-1:111111111111:function:FunctionName",
      "Condition": {
        "ArnLike": {
          "AWS:SourceArn": "arn:aws:iot:us-east-1:111111111111:authorizer/
AuthorizerName"
        },
        "StringEquals": {
          "AWS:SourceAccount": "111111111111"
        }
      }
    }
  ]
}
```

c. Esta política concede ao`InvokeFunction`permissão em sua função para oAWS IoT Corediretor. Se você não vir, você terá que adicioná-lo usando aAddPermissionAPI. O exemplo a seguir mostra como fazer isso usando oAWS CLI.

```
aws lambda add-permission --function-name FunctionName --principal
 iot.amazonaws.com --source-arn AuthorizerARn --statement-id Id-123 --action
 "lambda:InvokeFunction"
```

4. Se você vir invocações, verifique se não há erros. Um erro pode indicar que a função do Lambda não está lidando corretamente com o evento de conexão queAWS IoT Coreenvia para ele.

   Para obter informações sobre como lidar com o evento em sua função do Lambda, consultethe section called “Definir sua função do Lambda” (p. 321). É possível usar o recurso de teste naAWS LambdaConsole do (https://console.aws.amazon.com/lambda/) para codificar valores de teste na função para garantir que a função esteja lidando com eventos corretamente.

5. Se você vir invocações sem erros, mas seus dispositivos não conseguem se conectar (ou publicar, assinar e receber mensagens), o problema pode ser que a política retornada por sua função do Lambda não dá permissões para as ações que seus dispositivos estão tentando executar. Execute as etapas a seguir para determinar se algo está errado com a política retornada pela função.

   a. Usar uma Amazon CloudWatch Registra a consulta do Insights para verificar logs por um curto período de tempo para verificar se há falhas. A consulta de exemplo a seguir classifica eventos por carimbo de data/hora e procura falhas.

```
display clientId, eventType, status, @timestamp | sort @timestamp desc | filter
 status = "Failure"
```

   b. Atualize sua função do Lambda para registrar os dados aos quais ela está retornandoAWS IoT Coree o evento que aciona a função. Você pode usar esses logs para inspecionar a política que a função cria.

Investigar problemas de dispositivos

Se você não encontrar problemas com a invocação da função do Lambda ou com a política que a função retorna, procure problemas com as tentativas de conexão dos seus dispositivos. Solicitações de conexão malformadas podem causarAWS IoT CorePara não acionar seu autorizador. Problemas de conexão podem ocorrer nas camadas TLS e na aplicação.

Possíveis problemas na camada TLS:

- Os clientes devem passar um cabeçalho de nome de host (HTTP, MQTT sobre WebSockets) ou a extensão TLS de Indicação de Nome do Servidor (HTTP, MQTT sobre WebSockets, MQTT) em todas as solicitações de autenticação personalizadas. Em ambos os casos, o valor passado deve corresponder a uma das suas contasAWS IoT Coreendpoints de dados. Esses são os endpoints que são retornados quando você executa os seguintes comandos da CLI.
  - `aws iot describe-endpoint –endpoint type iot:data-ats`
  - `aws iot describe-endpoint –endpoint type`(para legado) VeriSign endpoints)
- Os dispositivos que usam autenticação personalizada para conexões MQTT também devem passar pela extensão do TLS, Application Layer Protocol Negotiation (ALPN), com um valor de`mqtt`.
- Atualmente, a autenticação personalizada está disponível apenas na porta 443.

Possíveis problemas na camada de aplicativo:

- Se a assinatura estiver ativada (`osigningDisabled`campo é falso em seu autorizador), procure os seguintes problemas de assinatura.
  - Certifique-se de que você esteja passando a assinatura de token no`x-amz-customauthorizer-signature`Cabeçalho ou em um parâmetro de string de consulta.
  - Certifique-se de que o serviço não esteja assinando um valor diferente do token.
  - Certifique-se de passar o token no cabeçalho ou parâmetro de consulta especificado na`token-key-name`campo em seu autorizador.
- Certifique-se de que o nome do autorizador que você passa no`x-amz-customauthorizer-name`parâmetro de string de cabeçalho ou consulta é válido ou que você tenha um autorizador padrão definido para sua conta.

Autorização

Autorização é o processo de concessão de permissões a uma identidade autenticada. Você concede permissões noAWS IoT CoreusandoAWS IoT Coree políticas do IAM. Este tópico aborda políticas do AWS IoT Core. Para obter mais informações sobre políticas do IAM, consulteIdentity and Access Management para o AWS IoT (p. 383)eComo o AWS IoT funciona com o IAM (p. 388).

As políticas do AWS IoT Core determinam o que uma identidade autenticada pode fazer. A identidade autenticada é usada por dispositivos, aplicativos móveis, aplicativos web e aplicativos de desktop. Uma identidade autenticada pode até mesmo ser um usuário digitando comandos da CLI do AWS IoT Core. Uma identidade poderá executar operações do AWS IoT Core somente se ela tiver uma política que conceda permissão para essas operações.

AmbosAWS IoT CoreAs políticas do IAM e políticas do IAM são usadas comAWS IoT CoreComo controlar as operações de uma identidade (também chamada dediretor) pode executar. O tipo de política usada depende do tipo de identidade que você está usando para fazer a autenticação com o AWS IoT Core.

As operações do AWS IoT Core são divididas em dois grupos:

- A API de plano de controle permite executar tarefas administrativas, como criar ou atualizar certificados, coisas, regras, e assim por diante.

- A API de plano de dados permite enviar e receber dados do AWS IoT Core.

O tipo de política usada depende se você estiver usando a API de plano de controle ou de data plane.

A tabela a seguir mostra os tipos de identidade, os protocolos usados e os tipos de política que podem ser usados para a autorização.

API de plano de dados e tipos de política do AWS IoT Core

Mecanismo de protocolo e autenticação	SDK	Tipo de identidade	Tipo de política		
MQTT sobre TLS/TCP, autenticação mútua TLS (porta 8883 ou 443)† (p. 84))	SDK de dispositivo do AWS IoT	Certificados X.509	Política do AWS IoT Core		
MQTT sobre HTTPS/ WebSocket,AWSAutenticação SigV4 (porta 443)	AWSSDK, móvel	Identity autenticada Amazon Cognito	IAM eAWS IoT Corepolíticas		
		Identity Amazon Cognito não autenticada	Política do IAM		
		IAM ou identidade federada	Política do IAM		
HTTPS,AWSAutenticação Signature versão 4 (porta 443)	AWS CLI	Amazon Cognito, IAM ou identidade federada	Política do IAM		
HTTPS, autenticação mútua TLS (porta 8443)	Não há suporte para o SDK	Certificados X.509	Política do AWS IoT Core		
HTTPS sobre autenticação personalizada (Porta 443)	SDK de dispositivo do AWS IoT	Autorizador personalizado	Política do autorizador personalizado		

API de plano de controle e tipos de política do AWS IoT Core

Mecanismo de protocolo e autenticação	SDK	Tipo de identidade	Tipo de política		
HTTPSAWSAutenticação Signature	AWS CLI	Amazon Cognito Identity	Política do IAM		

Mecanismo de protocolo e autenticação	SDK	Tipo de identidade	Tipo de política		
versão 4 (porta 443)		IAM ou identidade federada	Política do IAM		

AWS IoT CoreAs políticas do são anexadas a certificados X.509 ou a identidades do Amazon Cognito. As políticas do IAM são anexadas a um usuário, grupo ou função do IAM. Se você usar oAWS IoTconsole ou oAWS IoT CoreCLI do para anexar a política (a um certificado ou ao Amazon Cognito Identity), você usa umAWS IoT Corepolítica. Caso contrário, você usará uma política do IAM.

A autorização com base em políticas é uma ferramenta poderosa. Ela oferece controle total sobre o que um dispositivo, um usuário ou um aplicativo pode fazer no AWS IoT Core. Por exemplo, considere que um dispositivo que se conecta ao AWS IoT Core com um certificado. Você pode permitir que o dispositivo acesse todos os tópicos MQTT ou pode restringir seu acesso a um único tópico. Em outro exemplo, considere que um usuário digite comandos CLI na linha de comando. Ao usar uma política, você pode conceder ou negar acesso a qualquer comando ou recurso do AWS IoT Core ao usuário. Você também pode controlar o acesso de um aplicativo aos recursos do AWS IoT Core.

As alterações feitas em uma política podem levar alguns minutos para que se tornem efetivas devido a comoAWS IoTarmazena em cache os documentos da política. Ou seja, pode levar alguns minutos para acessar um recurso que recebeu acesso recentemente, e um recurso pode ficar acessível por vários minutos após o acesso ter sido revogado.

Treinamento e certificação da AWS

Para obter informações sobre autorização emAWS IoT Core, pegue oMergulhe profundamente noAWS IoT CoreAutenticação e autorizaçãocurso sobre oAWSSite de treinamento e certificação.

Políticas do AWS IoT Core

As políticas do AWS IoT Core são documentos JSON. Elas seguem as mesmas convenções que as políticas do IAM.AWS IoT CoreA é compatível com as regras nomeadas tantas quantas as identidades possam fazer referência no mesmo documento de política. As políticas nomeadas são versionadas para que possam ser facilmente restabelecidas.

As políticas do AWS IoT Core permitem controlar o acesso ao plano de dados do AWS IoT Core. OAWS IoT Coreplano de dados consiste em operações que permitem que você se conecte aoAWS IoT CoreO agente de mensagens da, envie e receba mensagens MQTT e obtenha ou atualize o Device Shadow de uma coisa.

Uma política do AWS IoT Core é um documento JSON que contém uma ou mais instruções de política. Cada instrução contém:

- `Effect`, que especifica se a ação é permitida ou negada.
- `Action`, que especifica a ação que a política está permitindo ou negando.
- `Resource`, que especifica o recurso ou os recursos nos quais a ação é permitida ou negada.

As alterações feitas em uma política podem levar alguns minutos para que se tornem efetivas devido a comoAWS IoTarmazena em cache os documentos da política. Ou seja, pode levar alguns minutos para acessar um recurso que recebeu acesso recentemente, e um recurso pode ficar acessível por vários minutos após o acesso ter sido revogado.

Tópicos

- Ações de políticas do AWS IoT Core (p. 333)
- Recursos de ação do AWS IoT Core (p. 335)
- Variáveis de política do AWS IoT Core (p. 336)
- Prevenção contra o ataque “Confused deputy” em todos os serviços (p. 342)
- Exemplos de políticas do AWS IoT Core (p. 343)
- Autorização com identidades do Amazon Cognito (p. 373)

Ações de políticas do AWS IoT Core

As ações de política a seguir são definidas pelo AWS IoT Core:

Ações da política MQTT

`iot:Connect`

Representa a permissão para se conectar ao agente de mensagem do AWS IoT Core. A permissão `iot:Connect` é verificada sempre que uma solicitação `CONNECT` é enviada ao operador. O agente de mensagens não permite que dois clientes com o mesmo ID de cliente permaneçam conectados ao mesmo tempo. Depois que o segundo cliente se conectar, o agente fechará a conexão existente. Usar a`iot:Connect`Para garantir que apenas clientes autorizados usando um ID de cliente específico possam se conectar.

`iot:GetRetainedMessage`

Representa a permissão para obter o conteúdo de uma única mensagem retida. Mensagens retidas são as mensagens que foram publicadas com o sinalizador RETAIN definido e armazenado porAWS IoT Core. Para obter permissão para obter uma lista de todas as mensagens retidas da conta, consulte`iot:ListRetainedMessages` (p. 333).

`iot:ListRetainedMessages`

Representa a permissão para recuperar informações resumidas sobre as mensagens retidas da conta, mas não o conteúdo das mensagens. Mensagens retidas são as mensagens que foram publicadas com o sinalizador RETAIN definido e armazenado porAWS IoT Core. O ARN de recurso especificado para esta ação deve ser`*`. Para obter permissão para obter o conteúdo de uma única mensagem retida, consulte`iot:GetRetainedMessage` (p. 333).

`iot:Publish`

Representa a permissão para publicar um tópico MQTT. Essa permissão é verificada sempre que uma solicitação PUBLISH é enviada ao operador. Você pode usar isso para permitir que os clientes publiquem padrões de tópicos específicos.

> **Note**
>
> Para conceder a permissão `iot:Publish`, conceda também a permissão `iot:Connect`.

`iot:Receive`

Representa a permissão para receber uma mensagem do AWS IoT Core. O`iot:Receive`A permissão é confirmada sempre que uma mensagem é entregue a um cliente. Como essa permissão é verificada em cada entrega, você pode usá-la para revogar permissões para clientes que estejam inscritos em um tópico.

IOT:RetainPublish

Representa a permissão para publicar uma mensagem MQTT com o indicador RETAIN definido.

> **Note**
>
> Para conceder a permissão `iot:RetainPublish`, conceda também a permissão `iot:Publish`.

`iot:Subscribe`

Representa a permissão para se inscrever em um filtro de tópico. Essa permissão é verificada sempre que uma solicitação SUBSCRIBE é enviada ao operador. Use-o para permitir que os clientes se inscrevam em tópicos que correspondam a padrões de tópicos específicos.

> **Note**
>
> Para conceder a permissão `iot:Subscribe`, conceda também a permissão `iot:Connect`.

Ações da política Device Shadow

`iot:DeleteThingShadow`

Representa a permissão para excluir o Device Shadow de uma coisa. O`iot:DeleteThingShadow`A permissão é verificada sempre que é feita uma solicitação para excluir o conteúdo do Device Shadow de uma coisa.

`iot:GetThingShadow`

Representa a permissão para recuperar o Device Shadow de uma coisa. O`iot:GetThingShadow`A permissão é verificada sempre que é feita uma solicitação para recuperar o conteúdo do Device Shadow de uma coisa.

`iot:ListNamedShadowsForThing`

Representa a permissão para listar uma coisa chamada Sombras. O`iot:ListNamedShadowsForThing`A permissão é verificada sempre que é feita uma solicitação para listar uma coisa chamada Shadows.

`iot:UpdateThingShadow`

Representa a permissão para atualizar uma shadow de dispositivo. O`iot:UpdateThingShadow`A permissão é verificada sempre que é feita uma solicitação para atualizar o conteúdo do Device Shadow de uma coisa.

> **Note**
>
> As ações da política de execução de trabalhos se aplicam apenas ao endpoint HTTP TLS. Se você usar o endpoint MQTT, deverá usar as ações de política do MQTT definidas neste tópico. Para obter um exemplo de uma política de execução de trabalhos que demonstra isso, consultethe section called “Exemplo de política de trabalho básica” (p. 372)O que funciona com o protocolo MQTT.

Ações de política do AWS IoT Core para execuções de trabalho

`iot:DescribeJobExecution`

Representa a permissão para recuperar uma execução de trabalho para uma determinada coisa. A permissão `iot:DescribeJobExecution` é verificada sempre que é feita uma solicitação para obter uma execução de tarefa.

`iot:GetPendingJobExecutions`

Representa a permissão para recuperar a lista de trabalhos que não estão em status terminal para uma coisa. A permissão `iot:GetPendingJobExecutions` é verificada sempre que é feita uma solicitação para recuperar a lista.

`iot:UpdateJobExecution`

Representa a permissão para atualizar uma execução de trabalho. A permissão `iot:UpdateJobExecution` é verificada sempre que é feita uma solicitação para atualizar o estado de uma execução de trabalho.

`iot:StartNextPendingJobExecution`

Representa a permissão para obter e iniciar a próxima execução de trabalho pendente para uma coisa. (Isto é, para atualizar uma execução de tarefa com status QUEUED para IN_PROGRESS). A permissão `iot:StartNextPendingJobExecution` é verificada sempre que é feita uma solicitação para iniciar a próxima execução de trabalho pendente.

AWS IoT CoreAção da política do provedor de credenciais

`iot:AssumeRoleWithCertificate`

Representa a permissão para ligarAWS IoT Coreprovedor de credenciais para assumir uma função do IAM com autenticação baseada em certificado. O`iot:AssumeRoleWithCertificate`A permissão é verificada sempre que é feita uma solicitação paraAWS IoT Coreprovedor de credenciais para assumir uma função.

Recursos de ação do AWS IoT Core

Para especificar um recurso para uma ação de política do AWS IoT Core, é necessário usar o ARN do recurso. Todos os ARNs dos recursos são da seguinte forma:

```
arn:aws:iot:region:AWS-account-ID:Resource-type/Resource-name
```

A tabela a seguir mostra o recurso a ser especificado para cada tipo de ação:

Ação	Tipo de recurso	Resource name (Nome do recurso)	Exemplo de ARN
`iot:Connect`	`client`	O ID do cliente	`arn:aws:iot:us-east-1:123456789012:client/myClientId`
`iot:DeleteThingShadow`	`thing`	O nome da coisa	`arn:aws:iot:us-east-1:123456789012:thing/thingOne`
`iot:DescribeJobExecution`	`thing`	O nome da coisa	`` `arn:aws:iot:us-east-1:123456789012:thing/thingOne` ``
`iot:GetPendingJobExecutions`	`thing`	O nome da coisa	`` `arn:aws:iot:us-east-1:123456789012:thing/thingOne` ``
`iot:GetRetainedMessage`	`topic`	Um tópico de mensagem retida.	`arn:aws:iot:us-east-1:123456789012:topic/myTopicName`
`iot:GetThingShadow`	`thing`	O nome da coisa	`arn:aws:iot:us-east-1:123456789012:thing/thingOne`
`iot:ListRetainedMessages`	`Todos`	Tudo	`*`
`iot:Publish`	`topic`	Uma string de tópicos	`arn:aws:iot:us-east-1:123456789012:topic/myTopicName`

Ação	Tipo de recurso	Resource name (Nome do recurso)	Exemplo de ARN
`iot:Receive`	`topic`	Uma string de tópicos	`arn:aws:iot:us-east-1:123456789012:topic/myTopicName`
`iot:RetainPublish`	`topic`	Um tópico a ser publicado com o sinalizador RETAIN definido.	`arn:aws:iot:us-east-1:123456789012:topic/myTopicName`
`iot:StartNextPendingJobExecution`	`thing`	O nome da coisa	`arn:aws:iot:us-east-1:123456789012:thing/thingOne`
`iot:Subscribe`	`topicfilter`	Uma string de filtro de tópicos	`arn:aws:iot:us-east-1:123456789012:topicfilt myTopicFilter`
`iot:UpdateJobExecution`	`thing`	O nome da coisa	`arn:aws:iot:us-east-1:123456789012:thing/thingOne`

`iot:UpdateThingShadow`	`thing`	O nome da coisa e o nome da sombra, se aplicável	`arn:aws:iot:us-east-1:123456789012:thing/thingOne`
			
			`arn:aws:iot:us-east-1:123456789012:thing/thingOne/shadowOne`

`iot:AssumeRoleWithCertificate`	`rolealias`	Um alias de função que aponta para um ARN de função	`arn:aws:iot:us-east-1:123456789012:rolealias CredentialProviderRole_alias`

Variáveis de política do AWS IoT Core

O AWS IoT Core define variáveis de política que podem ser usadas em políticas do AWS IoT Core no bloco `Resource` ou `Condition`. Quando uma política é avaliada, as variáveis da política são substituídas por valores reais. Por exemplo, se um dispositivo estiver conectado ao agente de mensagens do AWS IoT Core com um ID de cliente 100-234-3456, a variável de política `iot:ClientId` será substituída no documento da política por 100-234-3456.

AWS IoT Coreas políticas podem usar caracteres curinga e seguir uma convenção semelhante às políticas do IAM. Inserção de um`*`(asterik) na string pode ser tratado como um curinga, correspondendo a qualquer caractere. Por exemplo, você pode usar`*`para descrever vários nomes de tópicos do MQTT na`Resource`atributo de uma política. Os personagens`+`e`#`são tratados como cadeias de caracteres literais em uma política. Para obter um exemplo de política que mostra como usar curingas, consulte.Políticas para clientes MQTT (p. 351).

Também é possível usar variáveis de políticas predefinidas com valores fixos para representar caracteres que, de outra forma, têm um significado especial. Esses caracteres especiais incluem`$(*)`,`$(?)`, e`$($)`. Para obter mais informações sobre variáveis de política e os caracteres especiais, consulteElementos da política do IAM: Variáveis e tagseCriar uma condição com várias chaves ou valores.

Tópicos

- Variáveis básicas de política do AWS IoT Core (p. 337)

- Variáveis de política de coisas (p. 338)
- Variáveis de política do AWS IoT Core do certificado X.509 (p. 339)

Variáveis básicas de política do AWS IoT Core

O AWS IoT Core define as seguintes variáveis básicas de política:

- `iot:ClientId`: o ID do cliente usado para se conectar ao agente de mensagem do AWS IoT Core.
- `aws:SourceIp`: o endereço IP do cliente conectado ao agente de mensagem do AWS IoT Core.

A política do AWS IoT Core a seguir mostra uma política que usa variáveis de política:

```
{
    "Version": "2012-10-17",
    "Statement": [
      {
        "Effect": "Allow",
        "Action": ["iot:Connect"],
        "Resource": [
          "arn:aws:iot:us-east-1:123451234510:client/${iot:ClientId}"
        ]
      },
      {
        "Effect": "Allow",
        "Action": ["iot:Publish"],
        "Resource": [
          "arn:aws:iot:us-east-1:123451234510:topic/my/topic/${iot:ClientId}"
        ]
      }
    ]
}
```

Nesses exemplos, `${iot:ClientId}` é substituído pelo ID do cliente conectado ao agente de mensagem do AWS IoT Core quando a política é avaliada. Quando você usa variáveis de políticas, como `${iot:ClientId}`, você pode abrir acidentalmente o acesso a tópicos não intencionais. Por exemplo, se você usa uma política que usa `${iot:ClientId}` para especificar um filtro de tópico:

```
{
    "Effect": "Allow",
    "Action": ["iot:Subscribe"],
    "Resource": [
      "arn:aws:iot:us-east-1:123456789012:topicfilter/my/${iot:ClientId}/topic"
    ]
}
```

Um cliente pode se conectar usando + como o ID do cliente. Isso permitiria que o usuário se inscrevesse em qualquer tópico que correspondesse ao filtro de tópico `my/+/topic`. Para proteger contra essas falhas de segurança, use a ação da política `iot:Connect` para controlar quais IDs de cliente podem se conectar. Por exemplo, essa política permite que apenas clientes cujo ID é `clientid1` se conectem:

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

```
            "arn:aws:iot:us-east-1:123456789012:client/clientid1"
          ]
        }
      ]
}
```

Variáveis de política de coisas

As variáveis de política de coisas permitem escrever políticas do AWS IoT Core que concedam ou neguem permissões com base em propriedades de coisas, como nomes de coisas, tipos de coisas e valores de atributo de coisas. É possível usar variáveis de política de coisas a fim de aplicar a mesma política para controlar muitos dispositivos do AWS IoT Core. Para obter mais informações sobre o provisionamento de dispositivos, consulte Provisionamento de dispositivos. O nome da coisa é obtido do ID do cliente na mensagem MQTT `Connect` enviada quando uma coisa se conecta o AWS IoT Core.

Lembre-se do seguinte ao usar variáveis de política de coisas em políticas do AWS IoT Core.

- Usar aAttachThingPrincipalAPI para anexar certificados ou principais (identidades do Amazon Cognito autenticadas) a uma coisa.
- Quando você estiver substituindo nomes de coisas por variáveis de política de coisas, o valor de `clientId` na mensagem de conexão MQTT ou a conexão TLS deverá corresponder exatamente ao nome da coisa.

As variáveis de política de coisas a seguir estão disponíveis:

- `iot:Connection.Thing.ThingName`

  Isso resolve o nome da coisa no registro do AWS IoT Core para a qual a política está sendo avaliada. O AWS IoT Core usa o certificado que o dispositivo apresenta ao fazer a autenticação a fim de determinar qual coisa deve ser usada para verificar a conexão. Essa variável de política está disponível somente quando um dispositivo se conecta pelo MQTT ou MQTT pelo WebSocket protocolo.
- `iot:Connection.Thing.ThingTypeName`

  Isso resolve o tipo de coisa associado à coisa para a qual a política está sendo avaliada. O nome da coisa é definido como o ID do cliente da conexão MQTT/WebSocket. Essa variável de política está disponível somente na conexão pelo MQTT ou MQTT pelo protocolo WebSocket.
- `iot:Connection.Thing.Attributes[`*`attributeName`*`]`

  Isso resolve o valor do atributo especificado associado à coisa para a qual a política está sendo avaliada. Uma coisa pode ter até 50 atributos. Cada atributo está disponível como uma variável de política: `iot:Connection.Thing.Attributes[`*`attributeName`*`]` em que *`attributeName`* é o nome do atributo. O nome da coisa é definido como o ID do cliente da conexão MQTT/WebSocket. Essa variável de política está disponível somente na conexão pelo MQTT ou MQTT pelo WebSocket protocolo.
- `iot:Connection.Thing.IsAttached`

  `iot:Connection.Thing.IsAttached: ["true"]`impõe que apenas os dispositivos que estão registrados emAWS IoTe anexado ao principal pode acessar as permissões dentro da política. Você pode usar essa variável para evitar que um dispositivo se conecte aoAWS IoT Corese ele apresenta um certificado que não está anexado a uma coisa do IoT naAWS IoT CoreRegistr.Esta variável tem valores`true`ou`false`indicando que a coisa de conexão está anexada ao certificado ou à identidade do Amazon Cognito no registro usandoAttachThingPrincipalAPI. O nome da coisa é tomado como ID do cliente.

Variáveis de política do AWS IoT Core do certificado X.509

As variáveis de política do certificado X.509 permitem escrever políticas do AWS IoT Core que concedam permissões com base em atributos do certificado X.509. As seções a seguir descrevem como você pode usar essas variáveis de política de certificado.

CertificateId

Na API RegisterCertificate, o `certificateId` é exibido no corpo da resposta. Para obter informações sobre seu certificado, é possível usar `certificateId` DescribeCertificate.

Atributos de emissor

As variáveis de política do AWS IoT Core a seguir permitem conceder ou negar permissões com base em atributos de certificado definidos pelo emissor do certificado.

- `iot:Certificate.Issuer.DistinguishedNameQualifier`
- `iot:Certificate.Issuer.Country`
- `iot:Certificate.Issuer.Organization`
- `iot:Certificate.Issuer.OrganizationalUnit`
- `iot:Certificate.Issuer.State`
- `iot:Certificate.Issuer.CommonName`
- `iot:Certificate.Issuer.SerialNumber`
- `iot:Certificate.Issuer.Title`
- `iot:Certificate.Issuer.Surname`
- `iot:Certificate.Issuer.GivenName`
- `iot:Certificate.Issuer.Initials`
- `iot:Certificate.Issuer.Pseudonym`
- `iot:Certificate.Issuer.GenerationQualifier`

Atributos de assunto

As variáveis de política do AWS IoT Core a seguir permitem conceder ou negar permissões com base em atributos de assunto definidos pelo emissor do certificado.

- `iot:Certificate.Subject.DistinguishedNameQualifier`
- `iot:Certificate.Subject.Country`
- `iot:Certificate.Subject.Organization`
- `iot:Certificate.Subject.OrganizationalUnit`
- `iot:Certificate.Subject.State`
- `iot:Certificate.Subject.CommonName`
- `iot:Certificate.Subject.SerialNumber`
- `iot:Certificate.Subject.Title`
- `iot:Certificate.Subject.Surname`
- `iot:Certificate.Subject.GivenName`
- `iot:Certificate.Subject.Initials`
- `iot:Certificate.Subject.Pseudonym`
- `iot:Certificate.Subject.GenerationQualifier`

Os certificados X.509 permitem que esses atributos contenham um ou mais valores. Por padrão, as variáveis de política para cada atributo multivalor retornam o primeiro valor. Por exemplo, o atributo `Certificate.Subject.Country` pode conter uma lista de nomes de países, mas, quando avaliado em uma política, `iot:Certificate.Subject.Country` é substituído pelo primeiro nome do país. É possível solicitar um valor de atributo específico diferente do primeiro valor usando um índice baseado em um. Por exemplo, `iot:Certificate.Subject.Country.1` é substituído pelo segundo nome de país no atributo `Certificate.Subject.Country`. Se você especificar um valor de índice que não existe (por exemplo, se você solicitar um terceiro valor quando há apenas dois valores atribuídos ao atributo), nenhuma substituição será feita e haverá falha na autorização. Você pode usar o sufixo `.List` no nome da variável de política para especificar todos os valores do atributo.

Registered devices (2)

Para dispositivos registrados como coisas no registro do AWS IoT Core, a política a seguir permite que clientes com um nome de coisa registrado no registro do AWS IoT Core se conectem, mas restringe o direito de publicar um tópico específico do nome de coisa a clientes com certificados cujo atributo `Certificate.Subject.Organization` é definido como `"Example Corp"` ou `"AnyCompany"`. Essa restrição é realizada usando um campo `"Condition"` que especifica uma condição que deve ser atendida para permitir a ação anterior. Nesse caso, a condição é que o atributo `Certificate.Subject.Organization` associado ao certificado inclua um dos valores listados:

```
{
    "Version":"2012-10-17",
    "Statement":[
      {
        "Effect":"Allow",
        "Action":[
          "iot:Connect"
        ],
        "Resource":[
          "arn:aws:iot:us-east-1:123456789012:client/${iot:Connection.Thing.ThingName}"
        ]
      },
      {
        "Effect":"Allow",
        "Action":[
          "iot:Publish"
        ],
        "Resource":[
          "arn:aws:iot:us-east-1:123456789012:topic/my/topic/
${iot:Connection.Thing.ThingName}"
        ],
        "Condition":{
          "ForAllValues:StringEquals":{
            "iot:Certificate.Subject.Organization.List":[
              "Example Corp",
              "AnyCompany"
            ]
          }
        }
      }
    ]
}
```

Unregistered devices (2)

Para dispositivos não registrados como coisas no registro do AWS IoT Core, a política a seguir concede permissão para se conectar ao AWS IoT Core com os IDs de cliente `client1`, `client2` e `client3`, mas restringe o direito de publicar em um tópico específico de ID de cliente para os clientes com certificados cujo atributo `Certificate.Subject.Organization` esteja definido como `"Example Corp"` ou `"AnyCompany"`. Essa restrição é realizada usando um campo `"Condition"`

que especifica uma condição que deve ser atendida para permitir a ação anterior. Nesse caso, a condição é que o atributo `Certificate.Subject.Organization` associado ao certificado inclua um dos valores listados:

```
{
    "Version":"2012-10-17",
    "Statement":[
      {
        "Effect":"Allow",
        "Action":[
          "iot:Connect"
        ],
        "Resource":[
          "arn:aws:iot:us-east-1:123456789012:client/client1",
          "arn:aws:iot:us-east-1:123456789012:client/client2",
          "arn:aws:iot:us-east-1:123456789012:client/client3"
        ]
      },
      {
        "Effect":"Allow",
        "Action":[
          "iot:Publish"
        ],
        "Resource":[
          "arn:aws:iot:us-east-1:123456789012:topic/my/topic/${iot:ClientId}"
        ],
        "Condition":{
          "ForAllValues:StringEquals":{
            "iot:Certificate.Subject.Organization.List":[
              "Example Corp",
              "AnyCompany"
            ]
          }
        }
      }
    ]
}
```

Atributos de nome alternativo do emissor

As variáveis de política do AWS IoT Core a seguir permitem conceder ou negar permissões com base em atributos de nome alternativo do emissor definidos pelo emissor do certificado.

- `iot:Certificate.Issuer.AlternativeName.RFC822Name`
- `iot:Certificate.Issuer.AlternativeName.DNSName`
- `iot:Certificate.Issuer.AlternativeName.DirectoryName`
- `iot:Certificate.Issuer.AlternativeName.UniformResourceIdentifier`
- `iot:Certificate.Issuer.AlternativeName.IPAddress`

Atributos de nome alternativo do assunto

As variáveis de política do AWS IoT Core a seguir permitem conceder ou negar permissões com base em atributos de nome alternativo do assunto definidos pelo emissor do certificado.

- `iot:Certificate.Subject.AlternativeName.RFC822Name`
- `iot:Certificate.Subject.AlternativeName.DNSName`
- `iot:Certificate.Subject.AlternativeName.DirectoryName`
- `iot:Certificate.Subject.AlternativeName.UniformResourceIdentifier`

- `iot:Certificate.Subject.AlternativeName.IPAddress`

Outros atributos

É possível usar `iot:Certificate.SerialNumber` para permitir ou negar o acesso a recursos do AWS IoT Core com base no número de série de um certificado. A variável de política `iot:Certificate.AvailableKeys` contém o nome de todas as variáveis de política de certificado que contenham valores.

Limitações das variáveis de política de certificado X.509

As limitações a seguir se aplicam às variáveis de política de certificado X.509:

Curingas

Se caracteres curinga estiverem presentes em atributos de certificado, a variável de política não será substituída pelo valor do atributo do certificado, deixando o texto `${policy-variable}` no documento da política. Isso pode causar falha de autorização. Os seguintes caracteres curinga podem ser usados: `*`, `$`, `+`, `?` e `#`.

Campos da matriz

Os atributos de certificado que contêm matrizes estão limitados a cinco itens. Os itens adicionais são ignorados.

Tamanho da string

Todos os valores de string são limitados a 1024 caracteres. Se um atributo de certificado tiver uma sequência com mais de 1024 caracteres, a variável de política não será substituída pelo valor do atributo do certificado, deixando o texto `${policy-variable}` no documento da política. Isso pode causar falha de autorização.

Caracteres especiais

Qualquer caractere especial, como `,`, `"`, `\`, `+`, `=`, `<`, `>` e `;`, deve ser prefixado com uma barra invertida (`\`) quando usado em uma variável de política. Por exemplo, `Amazon Web Services O=Amazon.com Inc. L=Seattle ST=Washington C=US` torna-se `Amazon Web Service O \=Amazon.com Inc. L\=Seattle ST\=Washington C\=US`.

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.

Para limitar as permissões queAWS IoTfornece outro serviço ao recurso, recomendamos usar o`aws:SourceArn`e`aws:SourceAccount`Chaves de contexto de condição globais em políticas de recursos. 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 do deputado confuso é usar o`aws:SourceArn`Chave de contexto de condição global com o nome de recurso da Amazon (ARN)

completo do recurso. para oAWS IoT, seu`aws:SourceArn`Deve estar em conformidade com o formato: `arn:aws:iot:`*`region`*`:`*`account-id`*`:*`. Verifique se o*`região`*corresponde à suaAWS IoTRegião e o*`account-id`*O corresponde ao ID da sua conta de cliente.

O exemplo a seguir mostra como evitar o problema do deputado confuso usando o`aws:SourceArn`e`aws:SourceAccount`Chaves de contexto de condição globais naAWS IoTPolítica de confiança de função.

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

Exemplos de políticas do AWS IoT Core

Os exemplos de políticas nesta seção ilustram os documentos de políticas usados para concluir tarefas comuns noAWS IoT Core. Você pode usá-los como exemplos para começar ao criar as políticas para suas soluções.

Os exemplos desta seção usam esses elementos de política:

- the section called “Ações de políticas do AWS IoT Core” (p. 333)
- the section called “Recursos de ação do AWS IoT Core” (p. 335)
- the section called “Exemplos de políticas baseadas em identidade” (p. 404)
- the section called “Variáveis básicas de política do AWS IoT Core” (p. 337)
- the section called “Variáveis de política do AWS IoT Core do certificado X.509” (p. 339)

Exemplos de políticas nesta seção:

- Exemplos de política de conexão (p. 343)
- Exemplos de política de publicação/inscrição (p. 350)
- Exemplos de política para conectar e publicar (p. 365)
- Exemplos de política de mensagens retidas (p. 365)
- Exemplos de política de certificado (p. 367)
- Exemplos de política de coisas (p. 371)
- Exemplo de política de trabalho básica (p. 372)

Exemplos de política de conexão

A política a seguir concede permissão para conectar-se ao AWS IoT Core com o ID de cliente `client1`:

```
{
    "Version": "2012-10-17",
    "Statement": [
      {
        "Effect": "Allow",
        "Action": [
          "iot:Connect"
        ],
        "Resource": [
          "arn:aws:iot:us-east-1:123456789012:client/client1"
        ]
      }
    ]
}
```

A política a seguir nega permissão aos IDs de cliente `client1` e `client2` para se conectarem ao AWS IoT Core, permitindo que os dispositivos se conectem usando um ID de cliente que corresponda ao nome de uma coisa registrada no registro do AWS IoT Core.

```
{
    "Version": "2012-10-17",
    "Statement": [
      {
        "Effect": "Deny",
        "Action": [
          "iot:Connect"
        ],
        "Resource": [
          "arn:aws:iot:us-east-1:123456789012:client/client1",
          "arn:aws:iot:us-east-1:123456789012:client/client2"
        ]
      },
      {
        "Effect": "Allow",
        "Action": [
          "iot:Connect"
        ],
        "Resource": [
          "arn:aws:iot:us-east-1:123456789012:client/${iot:Connection.Thing.ThingName}"
        ]
      }
    ]
}
```

Exemplos de políticas de sessões persistentes do MQTT

`connectAttributes`permite especificar quais atributos você deseja usar em sua mensagem de conexão em suas políticas do IAM, como`PersistentConnecte`LastWill`. Para obter mais informações, consulteUsar ConnectAttribut (p. 95)

A política a seguir permite se conectar com`PersistentConnect`recurso:

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

```
                "ForAllValues:StringEquals": {
                    "iot:ConnectAttributes": [
                        "PersistentConnect"
                    ]
                }
            }
        }
    ]
}
```

A política a seguir não permite`PersistentConnect`, outros recursos são permitidos:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:Connect"
            ],
            "Resource": "*",
            "Condition": {
                "ForAllValues:StringNotEquals": {
                    "iot:ConnectAttributes": [
                        "PersistentConnect"
                    ]
                }
            }
        }
    ]
```

A política acima também pode ser expressa usando`StringEquals`, qualquer outro recurso, incluindo novo recurso, é permitido:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:Connect"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Deny",
            "Action": [
                "iot:Connect"
            ],
            "Resource": "*",
            "Condition": {
                "ForAnyValue:StringEquals": {
                    "iot:ConnectAttributes": [
                        "PersistentConnect"
                    ]
                }
            }
        }
    ]
}
```

A política a seguir permite se conectar por ambos`PersistentConnect`e`LastWill`, qualquer outro novo recurso não é permitido:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:Connect"
            ],
            "Resource": "*",
            "Condition": {
                "ForAllValues:StringEquals": {
                    "iot:ConnectAttributes": [
                        "PersistentConnect",
                        "LastWill"
                    ]
                }
            }
        }
    ]
}
```

A política a seguir permite uma conexão limpa por clientes com ou sem`LastWill`, nenhum outro recurso será permitido:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:Connect"
            ],
            "Resource": "*",
            "Condition": {
                "ForAllValues:StringEquals": {
                    "iot:ConnectAttributes": [
                        "LastWill"
                    ]
                }
            }
        }
    ]
}
```

A política a seguir só permite a conexão usando recursos padrão:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:Connect"
            ],
            "Resource": "*",
            "Condition": {
                "ForAllValues:StringEquals": {
                    "iot:ConnectAttributes": [
                    ]
                }
            }
        }
```

```
    ]
}
```

A política a seguir permite conectar-se somente com`PersistentConnect`, qualquer novo recurso é permitido desde que a conexão use`PersistentConnect`:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:Connect"
            ],
            "Resource": "*",
            "Condition": {
                "ForAnyValue:StringEquals": {
                    "iot:ConnectAttributes": [
                        "PersistentConnect"
                    ]
                }
            }
        }
    ]
}
```

A política a seguir afirma que a conexão deve ter ambos`PersistentConnect`e`LastWill`uso, nenhum novo recurso é permitido:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:Connect"
            ],
            "Resource": "*",
            "Condition": {
                "ForAllValues:StringEquals": {
                    "iot:ConnectAttributes": [
                        "PersistentConnect",
                        "LastWill"
                    ]
                }
            }
        },
        {
            "Effect": "Deny",
            "Action": [
                "iot:Connect"
            ],
            "Resource": "*",
            "Condition": {
                "ForAllValues:StringEquals": {
                    "iot:ConnectAttributes": [
                        "PersistentConnect"
                    ]
                }
            }
        },
        {
            "Effect": "Deny",
```

```
            "Action": [
                "iot:Connect"
            ],
            "Resource": "*",
            "Condition": {
                "ForAllValues:StringEquals": {
                    "iot:ConnectAttributes": [
                        "LastWill"
                    ]
                }
            }
        },
        {
            "Effect": "Deny",
            "Action": [
                "iot:Connect"
            ],
            "Resource": "*",
            "Condition": {
                "ForAllValues:StringEquals": {
                    "iot:ConnectAttributes": [
                    ]
                }
            }
        }
    ]
}
```

A seguinte política não deve ter`PersistentConnect`mas pode ter`LastWill`, qualquer outro novo recurso não é permitido:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Action": [
                "iot:Connect"
            ],
            "Resource": "*",
            "Condition": {
                "ForAnyValue:StringEquals": {
                    "iot:ConnectAttributes": [
                        "PersistentConnect"
                    ]
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "iot:Connect"
            ],
            "Resource": "*",
            "Condition": {
                "ForAllValues:StringEquals": {
                    "iot:ConnectAttributes": [
                        "LastWill"
                    ]
                }
            }
        }
    ]
}
```

A política a seguir permite a conexão somente por clientes que têm um`LastWill`Com tópico`"my/lastwill/topicName"`, qualquer recurso é permitido, desde que ele use a`LastWill`tópico:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:Connect"
            ],
            "Resource": "*",
            "Condition": {
                "ArnEquals": {
                    "iot:LastWillTopic": "arn:aws:iot:*region*:*account-id*:topic/*my/
lastwill/topicName*"
                }
            }
        }
    ]
}
```

A política a seguir só permite conexão limpa usando um específico`LastWillTopic`, qualquer recurso é permitido, desde que ele use a`LastWillTopic`:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:Connect"
            ],
            "Resource": "*",
            "Condition": {
                "ArnEquals": {
                    "iot:LastWillTopic": "arn:aws:iot:*region*:*account-id*:topic/*my/
lastwill/topicName*"
                }
            }
        },
        {
            "Effect": "Deny",
            "Action": [
                "iot:Connect"
            ],
            "Resource": "*",
            "Condition": {
                "ForAnyValue:StringEquals": {
                    "iot:ConnectAttributes": [
                        "PersistentConnect"
                    ]
                }
            }
        }
    ]
}
```

Registered devices (3)

A política a seguir concede permissão para um dispositivo se conectar usando o nome do item como o ID do cliente e para se inscrever no filtro de tópico `my/topic/filter`. O dispositivo deve ser

registrado no AWS IoT Core. Quando o dispositivo se conectar ao AWS IoT Core, ele deverá fornecer o certificado associado à coisa do IoT no registro do AWS IoT Core:

```
{
    "Version": "2012-10-17",
    "Statement": [
      {
        "Effect": "Allow",
        "Action": [
          "iot:Connect"
        ],
        "Resource": [
          "arn:aws:iot:us-east-1:123456789012:client/${iot:Connection.Thing.ThingName}"
        ]
      },
      {
        "Effect": "Allow",
        "Action": [
          "iot:Subscribe"
        ],
        "Resource": [
          "arn:aws:iot:us-east-1:123456789012:topicfilter/my/topic/filter"
        ]
      }
    ]
}
```

Unregistered devices (3)

Para dispositivos não registrados como coisas no registro do AWS IoT Core, a política a seguir concede permissão para se conectar usando o ID de cliente `client1` e inscrever-se no filtro de tópico `my/topic`:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:Connect"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:client/client1"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iot:Subscribe"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:topicfilter/my/topic"
            ]
        }
    ]
}
```

Exemplos de política de publicação/inscrição

A política usada depende de como você está se conectando aoAWS IoT Core. Você pode se conectar aAWS IoT CoreUsar um cliente MQTT, o HTTP ou WebSocket. Quando se conecta a um cliente MQTT,

você está se autenticando com um certificado X.509. Quando você se conecta por HTTP ou o WebSocket , você está se autenticando com o Signature versão 4 e o Amazon Cognito.

Políticas para clientes MQTT

Para descrever vários nomes de tópicos e filtros de tópicos na`resource`atributo de uma política, use o`*`e`?`caracteres curinga no lugar dos caracteres curinga MQTT.

MQTT eAWS IoT Coreas políticas têm caracteres curinga diferentes e devem ser escolhidas após uma cuidadosa consideração. No MQTT, os caracteres curinga`+`e`#`são usados emFiltros de tópicos MQTTpara se inscrever em vários nomes de tópicos.AWS IoT CoreUsar políticas do`*`e`?`como caracteres curinga e siga as convenções dePolíticas do IAM. Em um documento de política, o`*`Representa qualquer combinação de caracteres e um ponto de interrogação`?`representa qualquer caractere único. Em documentos de política, os caracteres curinga MQTT,`+`e`#`são tratados como esses caracteres sem significado especial.

Ao escolher caracteres curinga para usar em um documento de política, considere que o`*`caractere não se limita a um único nível de tópico como o`+`caractere está em um filtro de tópico MQTT. Para ajudar a restringir uma especificação curinga a um único nível de filtro de tópico MQTT, considere usar vários`?`characters. Para obter mais informações sobre o uso de caracteres curinga em um recurso de política e mais exemplos do que eles correspondem, consulteUso de caracteres curinga em ARNs de recursos.

A tabela a seguir mostra os diferentes caracteres curinga usados no MQTT eAWS IoT CorePolíticas para clientes MQTT.

Caractere curinga	É um caractere curinga MQTT	Exemplo no MQTT	ÉAWS IoT Corecaractere curinga da política	Exemplo emAWS IoT CorePolíticas para clientes MQTT
`#`	Sim	`some/#`	Não	N/D
`+`	Sim	`some/+/topic`	Não	N/D
`*`	Não	N/D	Sim	`topicfilter/some/*/topic`

`?`	Não	N/D	Sim	`topic/some/?????/topic`
				
				`topicfilter/some/sensor???/topic`

A política a seguir permite que um dispositivo publique em todos os subtópicos que começam com o mesmo nome de coisa.

Registered devices (5)

Para dispositivos registrados como coisas no registro do AWS IoT Core, a política a seguir concede permissão para se conectar ao AWS IoT Core usando um ID de cliente que corresponda ao nome da coisa e para publicar em qualquer tópico prefixado pelo nome da coisa:

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

```
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:client/
${iot:Connection.Thing.ThingName}"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iot:Publish"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:topic/
${iot:Connection.Thing.ThingName}/*"
            ]
        }
    ]
}
```

Unregistered devices (5)

Para dispositivos não registrados como coisas no registro do AWS IoT Core, a política a seguir concede permissão para conectar-se ao AWS IoT Core usando o ID de cliente `client1`, `client2` ou `client3` e para publicar em qualquer tópico prefixado pelo ID do cliente:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:Connect"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:client/client1",
                "arn:aws:iot:us-east-1:123456789012:client/client2",
                "arn:aws:iot:us-east-1:123456789012:client/client3"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iot:Publish"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:topic/${iot:ClientId}/*"
            ]
        }
    ]
}
```

Você também pode usar o caractere curinga * no final de um filtro de tópico. Como o uso de caracteres curinga pode levar à concessão não intencional de privilégios, eles só devem ser usados após uma avaliação cuidadosa. Uma situação na qual eles podem ser úteis é quando os dispositivos precisam se inscrever em mensagens com muitos tópicos diferentes (por exemplo, se um dispositivo deve se inscrever em relatórios de sensores de temperatura em vários locais).

Registered devices (6)

Para dispositivos registrados como coisas no registro do AWS IoT Core, a política a seguir concede permissão para conectar-se ao AWS IoT Core usando o nome do dispositivo como o ID do cliente e para inscrever-se em um tópico prefixado pelo nome da coisa, seguido por `room`, seguido por

qualquer string. (Espera-se que esses tópicos sejam, por exemplo, `thing1/room1`, `thing1/room2` e assim por diante):

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:Connect"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:client/
${iot:Connection.Thing.ThingName}"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iot:Subscribe"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:topicfilter/
${iot:Connection.Thing.ThingName}/room*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iot:Receive"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:topic/
${iot:Connection.Thing.ThingName}/room*"
            ]
        }
    ]
}
```

Unregistered devices (6)

Para dispositivos não registrados como coisas no registro do AWS IoT Core, a política a seguir concede permissão para conectar-se ao AWS IoT Core usando os IDs do cliente `client1`, `client2`, `client3` e para assinar um tópico prefixado pelo ID do cliente, seguido de `room`, seguido por qualquer string. (Espera-se que esses tópicos sejam, por exemplo, `client1/room1`, `client1/room2` e assim por diante):

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:Connect"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:client/client1",
                "arn:aws:iot:us-east-1:123456789012:client/client2",
                "arn:aws:iot:us-east-1:123456789012:client/client3"
            ]
        },
        {
            "Effect": "Allow",
```

```
            "Action": [
                "iot:Subscribe"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:topicfilter/${iot:ClientId}/room*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iot:Receive"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:topic/${iot:ClientId}/room*"
            ]
        }
    ]
}
```

Quando você especifica filtros de tópico emAWS IoT Corepolíticas para clientes MQTT, caracteres curinga MQTT+e#são tratados como cordas literais. Seu uso pode resultar em comportamento inesperado. Como elas são tratadas como cadeias de caracteres literais no momento da aplicação, existe a possibilidade de permitir o recebimento de dados involuntariamente por meio de um tópico curinga, mesmo se a política Negar estiver presente. Por exemplo, se um aplicativo quiser permitir que o dispositivo receba mensagens de todos os tópicos em (a/) exceto (a/restritado/...), o exemplo de política a seguir não funcionará:

```
{
     "Effect": "Allow",
     "Action": "iot:Subscribe",
     "Resource": "arn:aws:iot:us-east-1:123456789012:topicfilter/a/*"
 },
{
     "Effect": "Deny",
     "Action": "iot:Subscribe",
     "Resource": "arn:aws:iot:us-east-1:123456789012:topicfilter/a/restricted/*"
}
```

Com base no protocolo MQTT, assinando o filtro de tópicos`a/#`permitirá que os clientes recebam tópicos que incluem aqueles que começam com`a/restricted/[any string]`.

Para permitir que o dispositivo receba mensagens de todos os tópicos em (a/) exceto (a/restritado/...), você pode adicionar outra declaração:

```
{
   "Effect": "Allow",
   "Action": "iot:Receive",
   "NotResource": "arn:aws:iot:us-east-1:123456789012:topic/a/restricted/*"
}
```

Registered devices (4)

Para dispositivos registrados como coisas noAWS IoT CoreRegistro, a política a seguir concede permissão para se conectar aoAWS IoT CoreCom o ID do cliente que corresponde ao nome da coisa e para se inscrever no filtro de tópicos`some/+/topic`Somente . Observações no`topicfilter/some/+/topic`do recurso ARN, + é tratado como uma string literal emAWS IoT Corepolíticas para clientes MQTT, o que significa que somente a string`some/+/topic`Corresponde ao filtro de tópicos.

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

```
        {
            "Effect": "Allow",
            "Action": [
                "iot:Connect"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:client/
${iot:Connection.Thing.ThingName}"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iot:Subscribe"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:topicfilter/some/+/topic"
            ]
        }
    ]
}
```

Para dispositivos registrados como coisas noAWS IoT CoreRegistro, a política a seguir concede permissão para se conectar aoAWS IoT CoreCom o ID do cliente que corresponde ao nome da coisa e para se inscrever no filtro de tópicos`some/*/topic`. Observações no`topicfilter/some/*/topic`do ARN de recurso, * é tratado como um caractere curinga emAWS IoT Corepolíticas para clientes MQTT, o que significa que qualquer string no nível que contém o caractere corresponde ao filtro de tópico. (Espera-se que esses tópicos sejam, por exemplo,`some/string1/topic`,`some/string2/topic`, e assim por diante)

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:Connect"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:client/
${iot:Connection.Thing.ThingName}"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iot:Subscribe"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:topicfilter/some/*/topic"
            ]
        }
    ]
}
```

Unregistered devices (4)

Para dispositivos não registrados como coisas noAWS IoT CoreRegistro, a política a seguir concede permissão para se conectar aoAWS IoT CoreCom ID de clientes`client1`e se inscrever no filtro de tópicos`some/+/topic`Somente . Observações no`topicfilter/some/+/topic`do recurso ARN, + é tratado como uma string literal emAWS IoT Corepolíticas para clientes MQTT, o que significa que somente a string`some/+/topic`Corresponde ao filtro de tópicos.

```
{
    "Version": "2012-10-17",
    "Statement": [
      {
        "Effect": "Allow",
        "Action": [
          "iot:Connect"
        ],
        "Resource": [
          "arn:aws:iot:us-east-1:123456789012:client/client1"
        ]
      },
      {
        "Effect": "Allow",
        "Action": [
          "iot:Subscribe"
        ],
        "Resource": [
          "arn:aws:iot:us-east-1:123456789012:topicfilter/some/+/topic"
        ]
      }
    ]
}
```

Para dispositivos não registrados como coisas noAWS IoT CoreRegistro, a política a seguir concede permissão para se conectar aoAWS IoT CoreCom ID de clientes`client1`e se inscrever no filtro de tópicos`some/+/topic`. Observações no`topicfilter/some/*/topic`do ARN de recurso, * é tratado como um caractere curinga emAWS IoT Corepolíticas para clientes MQTT, o que significa que qualquer string no nível que contém o caractere corresponde ao filtro de tópico. (Espera-se que esses tópicos sejam, por exemplo,`some/string1/topic`,`some/string2/topic`, e assim por diante)

```
{
    "Version": "2012-10-17",
    "Statement": [
      {
        "Effect": "Allow",
        "Action": [
          "iot:Connect"
        ],
        "Resource": [
          "arn:aws:iot:us-east-1:123456789012:client/client1"
        ]
      },
      {
        "Effect": "Allow",
        "Action": [
          "iot:Subscribe"
        ],
        "Resource": [
          "arn:aws:iot:us-east-1:123456789012:topicfilter/some/*/topic"
        ]
      }
    ]
}
```

Note

Curingas MQTT+e#são tratados como cordas literais em umAWS IoT CorePolítica para clientes MQTT. Para especificar caracteres curinga em nomes de tópicos e filtros de tópicos emAWS IoT CorePolíticas para clientes MQTT, você deve usar`*`.

Registered devices (7)

Para dispositivos registrados como coisas no registro do AWS IoT Core, a política a seguir concede permissão para conectar-se ao AWS IoT Core usando o nome do dispositivo como o ID do cliente e para inscrever-se nos tópicos `my/topic` e `my/othertopic`:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:Connect"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:client/
${iot:Connection.Thing.ThingName}"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iot:Subscribe"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:topicfilter/my/topic",
                "arn:aws:iot:us-east-1:123456789012:topicfilter/my/othertopic"
            ]
        }
    ]
}
```

Unregistered devices (7)

Para dispositivos não registrados como coisas no registro do AWS IoT Core, a política a seguir concede permissão para conectar-se ao AWS IoT Core usando o ID de cliente `client1` e para inscrever-se nos tópicos `my/topic` e `my/othertopic`:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:Connect"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:client/client1"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iot:Subscribe"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:topicfilter/my/topic",
                "arn:aws:iot:us-east-1:123456789012:topicfilter/my/othertopic"
            ]
        }
    ]
}
```

Registered devices (8)

Para dispositivos registrados como coisas no registro do AWS IoT Core, a política a seguir concede permissão para conectar-se ao AWS IoT Core usando o nome do dispositivo como o ID do cliente e para inscrever-se em um tópico exclusivo para esse nome da coisa/ID do cliente:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:Connect"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:client/
${iot:Connection.Thing.ThingName}"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iot:Publish"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:topic/my/topic/
${iot:Thing.ThingName}"
            ]
        }
    ]
}
```

Unregistered devices (8)

Para dispositivos não registrados como coisas no registro do AWS IoT Core, a política a seguir concede permissão para conectar-se ao AWS IoT Core usando o ID de cliente `client1` e para publicar em um tópico exclusivo para esse ID de cliente:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:Connect"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:client/client1"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iot:Publish"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:topic/my/topic/${iot:ClientId}"
            ]
        }
    ]
}
```

Registered devices (9)

Para dispositivos registrados como coisas no registro do AWS IoT Core, a política a seguir concede permissão para se conectar ao AWS IoT Core usando o nome de coisa do dispositivo como o ID do cliente e para publicar em qualquer tópico prefixado por esse nome ou cliente, exceto para um tópico que termine com `bar`:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:Connect"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:client/
${iot:Connection.Thing.ThingName}"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iot:Publish"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:topic/${iot:Thing.ThingName}/*"
            ]
        },
        {
            "Effect": "Deny",
            "Action": [
                "iot:Publish"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:topic/${iot:Thing.ThingName}/bar"
            ]
        }
    ]
}
```

Unregistered devices (9)

Para dispositivos não registrados como coisas no registro do AWS IoT Core, a política a seguir concede permissão para se conectar ao AWS IoT Core usando os IDs de cliente `client1` e `client1` e para publicar em qualquer tópico prefixado pelo ID do cliente usado para conexão, exceto para um tópico que termine com `bar`:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:Connect"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:client/client1",
                "arn:aws:iot:us-east-1:123456789012:client/client2"
            ]
        },
        {
            "Effect": "Allow",
```

```
            "Action": [
                "iot:Publish"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:topic/${iot:ClientId}/*"
            ]
        },
        {
            "Effect": "Deny",
            "Action": [
                "iot:Publish"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:topic/${iot:ClientId}/bar"
            ]
        }
    ]
}
```

Registered devices (10)

Para dispositivos registrados como coisas no registro do AWS IoT Core, a política a seguir concede permissão para se conectar ao AWS IoT Core usando o nome da coisa do dispositivo como o ID do cliente. O dispositivo pode se inscrever no tópico`my/topic`, mas não pode publicar no*`thing-name`* `/bar`onde*`Nome da coisa`*É o nome da coisa da IoT se conectando aoAWS IoT Core:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:Connect"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:client/
${iot:Connection.Thing.ThingName}"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iot:Subscribe"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:topicfilter/my/topic"
            ]
        },
        {
            "Effect": "Deny",
            "Action": [
                "iot:Publish"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:topic/${iot:Thing.ThingName}/bar"
            ]
        }
    ]
}
```

Unregistered devices (10)

Para dispositivos não registrados como coisas no registro do AWS IoT Core, a política a seguir concede permissão para conectar-se ao AWS IoT Core usando o ID de cliente `client1` e para inscrever-se no tópico `my/topic`:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:Connect"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:client/client1"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iot:Subscribe"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:topicfilter/my/topic"
            ]
        }
    ]
}
```

As variáveis de política de coisas também são substituídas quando um certificado ou o Amazon Cognito Identity autenticado está anexado a uma coisa. A política a seguir concede permissão para se conectar ao AWS IoT Core com o ID de cliente `client1` e para publicar e receber no tópico `iotmonitor/provisioning/987654321098`. Também permite que o detentor de certificado se inscreva nesse tópico.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:Connect"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:client/client1"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iot:Publish",
                "iot:Receive"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:topic/iotmonitor/
provisioning/987654321098"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iot:Subscribe"
```

```
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:topicfilter/iotmonitor/
provisioning/987654321098"
            ]
        }
    ]
}
```

Políticas para clientes HTTP e WebSocket

As identidades do Amazon Cognito podem ser autenticadas ou não autenticadas. As identidades autenticadas pertencem a usuários que são autenticados por qualquer provedor de identidade. As identidades não autenticadas normalmente pertencem a usuários convidados que não se autenticam com um provedor de identidade. O Amazon Cognito fornece um identificador exclusivo eAWSCredenciais para suportar identidades não autenticadas.

Para as operações a seguir,AWS IoT CoreusaAWS IoT Corepolíticas anexadas às identidades do Amazon Cognito (por meio do`AttachPolicy`API) Para reduzir o escopo das permissões anexadas ao grupo de identidades do Amazon Cognito com identidades autenticadas.

- `iot:Connect`
- `iot:Publish`
- `iot:Subscribe`
- `iot:Receive`
- `iot:GetThingShadow`
- `iot:UpdateThingShadow`
- `iot:DeleteThingShadow`

Isso significa que uma Identidade do Amazon Cognito precisa da permissão da política de função do IAM anexada ao grupo e aoAWS IoT Corepolítica anexada ao Amazon Cognito Identity por meio doAWS IoT Core `AttachPolicy`API.

Usuários autenticados e não autenticados são tipos de identidade diferentes. Se você não anexar umAWS IoTpolítica para o Amazon Cognito Identity, um usuário autenticado falha na autorização emAWS IoTe não tem acesso aoAWS IoTRecursos e ações.

> **Note**
>
> Para outrosAWS IoT CoreOperações ou para identidades não autenticadas,AWS IoT CoreNão reduz o escopo das permissões anexadas à função de grupo de identidade do Amazon Cognito. Para as identidades autenticadas e não autenticadas, esta é a política mais permissiva que recomendamos anexar à função do grupo do Amazon Cognito.

HTTP

Para permitir que as identidades do Amazon Cognito não autenticadas publiquem mensagens via HTTP em um tópico específico para a identidade do Amazon Cognito, anexe a seguinte política do IAM à função do grupo de identidades do Amazon Cognito:

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

```
            "Resource": ["arn:aws:iot:us-east-1:123456789012:topic/${cognito-
identity.amazonaws.com:sub}"]
        }
    ]
}
```

Para permitir usuários autenticados, anexe a política anterior à função do conjunto de identidades do Amazon Cognito e à identidade do Amazon Cognito usando oAWS IoT Core AttachPolicyAPI.

> **Note**
>
> Ao autorizar identidades do Amazon Cognito,AWS IoT CoreO considera as duas políticas e concede os privilégios mínimos especificados. Uma ação será permitida somente se as duas políticas permitirem a ação solicitada. Se uma das políticas não permitir uma ação, essa ação não será autorizada.

MQTT

Para permitir que identidades não autenticadas do Amazon Cognito publiquem mensagens MQTT sobre WebSocket Em um tópico específico para a Amazon Cognito Identity em sua conta, anexe a seguinte política do IAM à função do grupo de identidades do Amazon Cognito:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:Publish"
            ],
            "Resource": ["arn:aws:iot:us-east-1:123456789012:topic/${cognito-
identity.amazonaws.com:sub}"]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iot:Connect"
            ],
            "Resource": ["arn:aws:iot:us-east-1:123456789012:client/${cognito-
identity.amazonaws.com:sub}"]
        }
    ]
}
```

Para permitir usuários autenticados, anexe a política anterior à função do conjunto de identidades do Amazon Cognito e à identidade do Amazon Cognito usando oAWS IoT Core AttachPolicyAPI.

> **Note**
>
> Ao autorizar identidades do Amazon Cognito,AWS IoT CoreConsidera ambos e concede os privilégios mínimos especificados. Uma ação será permitida somente se as duas políticas permitirem a ação solicitada. Se uma das políticas não permitir uma ação, essa ação não será autorizada.

Exemplos de política de recepção

Registered devices (11)

Para dispositivos registrados no registro do AWS IoT Core, a política a seguir concede permissão para se conectar ao AWS IoT Core com um ID de cliente que corresponda ao nome da coisa e para se inscrever no tópico `my/topic` e receber mensagens nele:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:Connect"
            ],
            "Resource": ["arn:aws:iot:us-east-1:123456789012:client/
${iot:Connection.Thing.ThingName}"]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iot:Subscribe"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:topicfilter/my/topic"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iot:Receive"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:topic/my/topic"
            ]
        }
    ]
}
```

Unregistered devices (11)

Para dispositivos não registrados no registro do AWS IoT Core, a política a seguir concede permissão para se conectar ao AWS IoT Core com o ID de cliente `client1` e para se inscrever em um tópico e receber mensagens nele:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:Connect"
            ],
            "Resource": ["arn:aws:iot:us-east-1:123456789012:client/client1"]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iot:Subscribe"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:topicfilter/my/topic"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iot:Receive"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:topic/my/topic"
```

```
            ]
        }
    ]
}
```

Exemplos de política para conectar e publicar

Para dispositivos registrados como coisas no registro do AWS IoT Core, a política a seguir concede permissão para se conectar ao AWS IoT Core com um ID de cliente que corresponda ao nome da coisa e restrinja o dispositivo à publicação em um tópico MQTT específico do ID de cliente ou nome de coisa. Para que uma conexão seja bem-sucedida, o nome da coisa deve ser registrado no registro do AWS IoT Core e ser autenticado usando uma identidade ou um principal anexado à coisa:

```
{
    "Version": "2012-10-17",
    "Statement": [
      {
        "Effect": "Allow",
        "Action":["iot:Publish"],
        "Resource": ["arn:aws:iot:us-east-1:123456789012:topic/
${iot:Connection.Thing.ThingName}"]
      },
      {
        "Effect": "Allow",
        "Action": ["iot:Connect"],
        "Resource": ["arn:aws:iot:us-east-1:123456789012:client/
${iot:Connection.Thing.ThingName}"]
      }
    ]
}
```

Para dispositivos não registrados como coisas no registro do AWS IoT Core, a política a seguir concede permissão para conectar-se ao AWS IoT Core com o ID de cliente `client1` e restringe o dispositivo a publicar em um tópico MQTT específico do clientID.

```
{
    "Version": "2012-10-17",
    "Statement": [
      {
        "Effect": "Allow",
        "Action":["iot:Publish"],
        "Resource": ["arn:aws:iot:us-east-1:123456789012:topic/${iot:ClientId}"]
      },
      {
        "Effect": "Allow",
        "Action": ["iot:Connect"],
        "Resource": ["arn:aws:iot:us-east-1:123456789012:client/client1"]
      }
    ]
}
```

Exemplos de política de mensagens retidas

O uso doMensagens retidas (p. 90)Requer políticas específicas. Mensagens retidas são mensagens MQTT publicadas com o sinalizador RETAIN definido e armazenado porAWS IoT Core. Esta seção apresenta exemplos de políticas que permitem o uso comum de mensagens retidas.

Nesta seção:

- Política para conectar e publicar mensagens retidas (p. 366)

- Política para conectar e publicar mensagens do Will retidas (p. 366)
- Política para listar e receber mensagens retidas (p. 367)

Política para conectar e publicar mensagens retidas

Para que um dispositivo publique mensagens retidas, o dispositivo deve ser capaz de se conectar, publicar (qualquer mensagem MQTT) e publicar mensagens retidas do MQTT. A política a seguir concede essas permissões para o tópico:`device/sample/configuration`para cliente**`device1`**. Para outro exemplo que concede permissão para se conectar, consulte.the section called “Exemplos de política para conectar e publicar” (p. 365).

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
    {
      "Effect": "Allow",
      "Action": [
        "iot:Connect"
      ],
      "Resource": [
        "arn:aws:iot:us-east-1:123456789012:client/device1"
      ]
    },
      "Effect": "Allow",
      "Action": [
        "iot:Publish",
        "iot:RetainPublish"
      ],
      "Resource": [
        "arn:aws:iot:us-east-1:123456789012:topic/device/sample/configuration"
      ]
    }
  ]
}
```

Política para conectar e publicar mensagens do Will retidas

Os clientes podem configurar uma mensagem queAWS IoT Corepublicará quando o cliente se desconectar inesperadamente. O MQTT chama essa mensagem deWillmensagem. Um cliente deve ter uma condição adicional adicionada à permissão de conexão para incluí-lo.

O documento de política a seguir concede a todos os clientes permissão para se conectar e publicar uma mensagem Will, identificada por seu tópico,`will`, queAWS IoT Coretambém reterá.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iot:Connect"
      ],
      "Resource": [
        "arn:aws:iot:us-east-1:123456789012:client/*"
      ],
      "Condition": {
        "ForAllValues:StringEquals": {
          "iot:ConnectAttributes": [
            "LastWill"
          ]
```

```
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": [
        "iot:Publish",
        "iot:RetainPublish"
      ],
      "Resource": [
        "arn:aws:iot:us-east-1:123456789012:topic/will"
      ]
    }
  ]
}
```

Política para listar e receber mensagens retidas

Serviços e aplicativos podem acessar mensagens retidas sem a necessidade de oferecer suporte a um cliente MQTT ligando`ListRetainedMessages`e`GetRetainedMessage`. Os serviços e aplicativos que chamam essas ações devem ser autorizados usando uma política como o exemplo a seguir.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iot:ListRetainedMessages"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "iot:GetRetainedMessage"
      ],
      "Resource": [
        "arn:aws:iot:us-east-1:123456789012:topic/foo"
      ]
    }
  ]
}
```

Exemplos de política de certificado

Para dispositivos registrados no registro do AWS IoT Core, a política a seguir concede permissão para se conectar ao AWS IoT Core com um ID de cliente que corresponda ao nome de uma coisa e para publicar em um tópico cujo nome seja igual ao `certificateId` do certificado que o dispositivo usou para se autenticar:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:Publish"
            ],
            "Resource": ["arn:aws:iot:us-east-1:123456789012:topic/${iot:CertificateId}"]
```

```
        },
        {
            "Effect": "Allow",
            "Action": [
                "iot:Connect"
            ],
            "Resource": ["arn:aws:iot:us-east-1:123456789012:client/
${iot:Connection.Thing.ThingName}"]
        }
    ]
}
```

Para dispositivos não registrados no registro do AWS IoT Core, a política a seguir concede permissão para se conectar ao AWS IoT Core com IDs de cliente `client1`, `client2` e `client3` e para publicar em um tópico cujo nome seja igual ao `certificateId` do certificado que o dispositivo usou para se autenticar:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:Publish"
            ],
            "Resource": ["arn:aws:iot:us-east-1:123456789012:topic/${iot:CertificateId}"]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iot:Connect"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:client/client1",
                "arn:aws:iot:us-east-1:123456789012:client/client2",
                "arn:aws:iot:us-east-1:123456789012:client/client3"
            ]
        }
    ]
}
```

Para dispositivos registrados no registro do AWS IoT Core, a política a seguir concede permissão para se conectar ao AWS IoT Core com um ID de cliente que corresponda ao nome da coisa e para publicar em um tópico cujo nome seja igual ao campo `CommonName` do assunto do certificado que o dispositivo usou para se autenticar:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:Publish"
            ],
            "Resource": ["arn:aws:iot:us-east-1:123456789012:topic/
${iot:Certificate.Subject.CommonName}"]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iot:Connect"
            ],
            "Resource": ["arn:aws:iot:us-east-1:123456789012:client/
${iot:Connection.Thing.ThingName}"]
```

```
        }
    ]
}
```

Note

Neste exemplo, o nome comum do assunto do certificado é usado como o identificador do tópico, supondo-se que o nome comum do assunto seja exclusivo para cada certificado registrado. Se os certificados forem compartilhados entre vários dispositivos, o nome comum do assunto será o mesmo para todos os dispositivos que compartilham esse certificado, permitindo assim privilégios de publicação para o mesmo tópico em vários dispositivos (não recomendado).

Para dispositivos não registrados no registro do AWS IoT Core, a política a seguir concede permissão para se conectar ao AWS IoT Core com IDs de cliente `client1`, `client2` e `client3` e para publicar em um tópico cujo nome seja igual ao campo `CommonName` do assunto do certificado que o dispositivo usou para se autenticar:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:Publish"
            ],
            "Resource": ["arn:aws:iot:us-east-1:123456789012:topic/
${iot:Certificate.Subject.CommonName}"]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iot:Connect"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:client/client1",
                "arn:aws:iot:us-east-1:123456789012:client/client2",
                "arn:aws:iot:us-east-1:123456789012:client/client3"
            ]
        }
    ]
}
```

Note

Neste exemplo, o nome comum do assunto do certificado é usado como o identificador do tópico, supondo-se que o nome comum do assunto seja exclusivo para cada certificado registrado. Se os certificados forem compartilhados entre vários dispositivos, o nome comum do assunto será o mesmo para todos os dispositivos que compartilham esse certificado, permitindo assim privilégios de publicação para o mesmo tópico em vários dispositivos (não recomendado).

Para dispositivos registrados no registro do AWS IoT Core, a política a seguir concede permissão para se conectar ao AWS IoT Core com um ID de cliente que corresponda ao nome da coisa e para publicar em um tópico cujo nome é prefixado com `admin/` quando o certificado usado para autenticar o dispositivo tiver seu campo `Subject.CommonName.2` definido como `Administrator`:

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

```
            "Action": [
                "iot:Connect"
            ],
            "Resource": ["arn:aws:iot:us-east-1:123456789012:client/
${iot:Connection.Thing.ThingName}"]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iot:Publish"
            ],
            "Resource": ["arn:aws:iot:us-east-1:123456789012:topic/admin/*"],
            "Condition": {
                "StringEquals": {
                    "iot:Certificate.Subject.CommonName.2": "Administrator"
                }
            }
        }
    ]
}
```

Para dispositivos não registrados no registro do AWS IoT Core, a política a seguir concede permissão para se conectar ao AWS IoT Core com IDs de cliente `client1`, `client2` e `client3` e para publicar em um tópico cujo nome é prefixado com `admin/` quando o certificado usado para autenticar o dispositivo tiver seu campo `Subject.CommonName.2` definido como `Administrator`:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:Connect"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:client/client1",
                "arn:aws:iot:us-east-1:123456789012:client/client2",
                "arn:aws:iot:us-east-1:123456789012:client/client3"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iot:Publish"
            ],
            "Resource": ["arn:aws:iot:us-east-1:123456789012:topic/admin/*"],
            "Condition": {
                "StringEquals": {
                    "iot:Certificate.Subject.CommonName.2": "Administrator"
                }
            }
        }
    ]
}
```

Para dispositivos registrados no registro do AWS IoT Core, a política a seguir permite que um dispositivo use um nome de coisa para publicar em um tópico específico que consiste em `admin/` seguido pelo `ThingName` quando o certificado usado para autenticar o dispositivo tiver qualquer um de seus campos `Subject.CommonName` definidos como `Administrator`:

```
{
    "Version": "2012-10-17",
```

```
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:Connect"
            ],
            "Resource": ["arn:aws:iot:us-east-1:123456789012:client/
${iot:Connection.Thing.ThingName}"]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iot:Publish"
            ],
            "Resource": ["arn:aws:iot:us-east-1:123456789012:topic/admin/
${iot:Connection.Thing.ThingName}"],
            "Condition": {
                "ForAnyValue:StringEquals": {
                    "iot:Certificate.Subject.CommonName.List": "Administrator"
                }
            }
        }
    ]
}
```

Para dispositivos não registrados no registro do AWS IoT Core, a política a seguir concede permissão para se conectar ao AWS IoT Core com IDs de cliente `client1`, `client2` e `client3` para publicar no tópico `admin` quando o certificado usado para autenticar o dispositivo tiver qualquer um dos seus campos `Subject.CommonName` definidos como `Administrator`:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:Connect"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:client/client1",
                "arn:aws:iot:us-east-1:123456789012:client/client2",
                "arn:aws:iot:us-east-1:123456789012:client/client3"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iot:Publish"
            ],
            "Resource": ["arn:aws:iot:us-east-1:123456789012:topic/admin"],
            "Condition": {
                "ForAnyValue:StringEquals": {
                    "iot:Certificate.Subject.CommonName.List": "Administrator"
                }
            }
        }
    ]
}
```

Exemplos de política de coisas

A política a seguir permite que um dispositivo seja conectado se o certificado usado para fazer a autenticação com o AWS IoT Core está anexado à coisa para a qual a política está sendo avaliada:

```
{
    "Version":"2012-10-17",
    "Statement":[
        {
            "Effect":"Allow",
            "Action":["iot:Connect"],
            "Resource":[ "*" ],
            "Condition": {
                "Bool": {
                    "iot:Connection.Thing.IsAttached": ["true"]
                }
            }
        }
    ]
}
```

Exemplo de política de trabalho básica

Este exemplo mostra os status de política necessários para um destino de trabalho que é um único dispositivo para receber uma solicitação de trabalho e comunicar o status de execução do trabalho comAWS IoT.

Substituir*us-west- 2:57 EXEMPLO833*Com suasRegião da AWS, um caractere de dois pontos (:) e seus 12 dígitosConta da AWSNúmero e, em seguida, substitua*UniqueThingName*com o nome do recurso coisa que representa o dispositivo emAWS IoT.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iot:Connect"
      ],
      "Resource": [
        "arn:aws:iot:us-west-2:57EXAMPLE833:client/uniqueThingName"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "iot:Publish"
      ],
      "Resource": [
        "arn:aws:iot:us-west-2:57EXAMPLE833:topic/test/dc/pubtopic",
        "arn:aws:iot:us-west-2:57EXAMPLE833:topic/$aws/events/job/*",
        "arn:aws:iot:us-west-2:57EXAMPLE833:topic/$aws/events/jobExecution/*",
        "arn:aws:iot:us-west-2:57EXAMPLE833:topic/$aws/things/uniqueThingName/jobs/*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "iot:Subscribe"
      ],
      "Resource": [
        "arn:aws:iot:us-west-2:57EXAMPLE833:topicfilter/test/dc/subtopic",
        "arn:aws:iot:us-west-2:57EXAMPLE833:topic/$aws/events/jobExecution/*",
        "arn:aws:iot:us-west-2:57EXAMPLE833:topicfilter/$aws/things/uniqueThingName/jobs/*"
      ]
    },
    {
      "Effect": "Allow",
```

```
      "Action": [
        "iot:Receive"
      ],
      "Resource": [
        "arn:aws:iot:us-west-2:57EXAMPLE833:topic/test/dc/subtopic",
        "arn:aws:iot:us-west-2:57EXAMPLE833:topic/$aws/things/uniqueThingName/jobs/*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "iot:DescribeJobExecution",
        "iot:GetPendingJobExecutions",
        "iot:StartNextPendingJobExecution",
        "iot:UpdateJobExecution"
      ],
      "Resource": [
        "arn:aws:iot:us-west-2:57EXAMPLE833:topic/$aws/things/uniqueThingName"
      ]
    }
  ]
}
```

Autorização com identidades do Amazon Cognito

Existem dois tipos de identidades do Amazon Cognito: autenticados e não autenticados. Se o aplicativo oferecer suporte a identidades do Amazon Cognito não autenticadas, nenhuma autenticação será executada para que você não saiba quem é o usuário. Para esses usuários, você concede permissão anexando uma função do IAM a um grupo de identidades não autenticadas. Recomendamos que você conceda acesso somente aos recursos que você deseja disponibilizar para usuários desconhecidos.

Quando o aplicativo oferecer suporte a identidades do Amazon Cognito autenticadas, para autenticar usuários, será necessário especificar uma política em dois locais. Anexe uma política do IAM ao pool autenticado do Amazon Cognito Identity e anexe umAWS IoT CorePolítica para o Amazon Cognito Identity.

Usuários autenticados e não autenticados são tipos de identidade diferentes. Se você não anexar umAWS IoTpolítica para o Amazon Cognito Identity, um usuário autenticado falha na autorização emAWS IoTe não tem acesso aoAWS IoTRecursos e ações. Para obter mais informações sobre como criar políticas para identidades do Amazon Cognito, consulteExemplos de política de publicação/inscrição (p. 350).

O exemplo a seguir aplicativos da Web em GitHub mostrar como incorporar o anexo de política a usuários autenticados no processo de inscrição e autenticação do usuário.

- MQTT publicar/assinar o aplicativo Web React usandoAWS AmplifyO e aAWS IoT Device SDK for JavaScript
- MQTT publicar/assinar o aplicativo Web React usandoAWS Amplify, oAWS IoT Device SDK for JavaScript, e uma função Lambda

O Amplify é um conjunto de ferramentas e serviços que ajudam você a criar aplicativos móveis e web que se integram comAWSServiços da . Para obter mais informações sobre o Amplify, consulteDocumentação do Amplify Framework,.

Ambos os exemplos executam as etapas a seguir.

1. Quando um usuário se inscreve em uma conta, o aplicativo cria um grupo de usuários e uma identidade do Amazon Cognito.
2. Quando um usuário se autentica, o aplicativo cria e anexa uma política à identidade. Isso dá ao usuário permissões de publicação e assinatura.
3. O usuário pode usar o aplicativo para publicar e assinar tópicos do MQTT.

O primeiro exemplo usa `oAttachPolicy`API diretamente dentro da operação de autenticação. O exemplo a seguir demonstra como implementar essa API dentro de um aplicativo da Web React que usa o Amplify e oAWS IoT Device SDK for JavaScript.

```
function attachPolicy(id, policyName) {
    var Iot = new AWS.Iot({region: AWSConfiguration.region, apiVersion:
 AWSConfiguration.apiVersion, endpoint: AWSConfiguration.endpoint});
    var params = {policyName: policyName, target: id};

    console.log("Attach IoT Policy: " + policyName + " with cognito identity id: " + id);
    Iot.attachPolicy(params, function(err, data) {
         if (err) {
               if (err.code !== 'ResourceAlreadyExistsException') {
                  console.log(err);
               }
          }
         else  {
            console.log("Successfully attached policy with the identity", data);
         }
     });
}
```

Esse código aparece noArquivo AuthDisplay.jsfile.

O segundo exemplo implementa `oAttachPolicy`API em uma função do Lambda. O exemplo a seguir mostra como o Lambda usa essa API.

```
iot.attachPolicy(params, function(err, data) {
     if (err) {
           if (err.code !== 'ResourceAlreadyExistsException') {
              console.log(err);
              res.json({error: err, url: req.url, body: req.body});
           }
      }
     else  {
        console.log(data);
        res.json({success: 'Create and attach policy call succeed!', url: req.url, body:
 req.body});
     }
 });
```

Esse código aparece dentro do`iot.GetPolicy`Função noapp.jsfile.

> **Note**
>
> Quando você chama a função comAWScredenciais que você obtém por meio dos pools de identidade do Amazon Cognito, o objeto de contexto em sua função do Lambda contém um valor para`context.cognito_identity_id`. Para obter mais informações, consulte:
>
> - AWS LambdaObjeto de contexto do em Node.js
> - AWS LambdaObjeto de contexto do em Python
> - AWS LambdaObjeto de contexto do em Ruby
> - AWS LambdaObjeto de contexto do em Java
> - AWS LambdaObjeto de contexto do em Go
> - AWS LambdaObjeto de contexto do em C#
> - AWS LambdaObjeto de contexto do no PowerShell

Autorizar chamadas diretas paraAWSUsar serviços doAWS IoT CoreProvedor de credenciais

Os dispositivos podem usar certificados X.509 para se conectar ao AWS IoT Core usando protocolos de autenticação mútua TLS. Other (Outros)AWSOs serviços não são compatíveis com autenticação baseada em certificado, mas podem ser chamados usandoAWSCredenciais do emAWSFormato Signature versão 4. OAlgoritmo Signature versão 4Normalmente, o chamador tem um ID de chave de acesso e uma chave de acesso secreta.AWS IoT CoreTem um provedor de credenciais que permite que você use o integradocertificado X.509como a identidade exclusiva do dispositivo a ser autenticadaAWSSolicitações. Isso elimina a necessidade de armazenar um ID de chave de acesso e uma chave de acesso secreta em seu dispositivo.

O provedor de credenciais autentica um chamador usando um certificado X.509 e emite um token de segurança temporário e de privilégio limitado. O token pode ser usado para assinar e autenticar qualquerAWSsolicitação. Esta forma de autenticar seuAWSsolicitações exigem que você crie e configure umAWS Identity and Access ManagementFunção (IAM) do IAM)E anexe políticas do IAM adequadas à função para que o provedor de credenciais possa assumir a função em seu nome. Para obter mais informações sobreAWS IoT Coree IAM, consulteIdentity and Access Management para o AWS IoT (p. 383).

O diagrama a seguir mostra o fluxo de trabalho do provedor de credenciais.

1. O dispositivo do AWS IoT Core faz uma solicitação HTTPS ao provedor de credenciais para receber um token de segurança. A solicitação inclui o dispositivo certificado X.509 para autenticação.
2. O provedor de credenciais encaminha a solicitação ao módulo de autenticação e autorização do AWS IoT Core para validar o certificado e verificar se o dispositivo tem permissão para solicitar o token de segurança.

3. Se o certificado for válido e tiver permissão para solicitar um token de segurança, o módulo de autenticação e autorização do AWS IoT Core retornará uma resposta de êxito. Do contrário, envia uma exceção ao dispositivo.
4. Depois de validar o certificado, o provedor de credenciais chama o AWS Security Token Service (AWS STS) para assumir a função do IAM que você criou para ele.
5. O AWS STS retorna um token de segurança temporário e de privilégio limitado para o provedor de credenciais.
6. O provedor de credenciais retorna o token de segurança para o dispositivo.
7. O dispositivo usa o token de segurança para assinar umAWSSolicitar com oAWSSignature versão 4.
8. O serviço solicitado chama o IAM para validar a assinatura e autorizar a solicitação comparativamente às políticas de acesso anexadas à função do IAM que você criou para o provedor de credenciais.
9. Se o IAM valida a assinatura e autoriza a solicitação, a solicitação tem êxito. Do contrário, o IAM envia uma exceção.

A seção a seguir descreve como se deve usar um certificado para obter um token de segurança. Ele é escrito com a suposição de que você já tenha registrado um dispositivo e criado e ativado seu próprio certificado para ele.

Como usar um certificado para obter um token de segurança

1. Configure a função do IAM que o provedor de credenciais assume em nome de seu dispositivo. Anexe à função as políticas de confiança a seguir.

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

   Para cadaAWSO serviço a ser chamado, anexe uma política de acesso à função. O provedor de credenciais comporta as seguintes variáveis de política:

   - `credentials-iot:ThingName`
   - `credentials-iot:ThingTypeName`
   - `credentials-iot:AwsCertificateId`

   Quando o dispositivo fornece o nome da coisa em sua solicitação a umAWSserviço, o provedor de credenciais adiciona`credentials-iot:ThingName`e`credentials-iot:ThingTypeName`como variáveis de contexto para o token de segurança. O provedor de credenciais fornece `credentials-iot:AwsCertificateId` como uma variável de contexto, mesmo que o dispositivo não forneça o nome da coisa na solicitação. Você passa o nome da coisa como o valor do cabeçalho da solicitação HTTP `x-amzn-iot-thingname`.

   Essas três variáveis funcionam apenas para políticas do IAM, e não para políticas da AWS IoT Core.
2. O usuário que executar a próxima etapa (criar um alias de função) deve ter permissão para transmitir a função recém-criada ao AWS IoT Core. A política a seguir fornece ambos`iam:GetRole`e`iam:PassRole`Permissões para umAWSusuário. A permissão `iam:GetRole` permite que o usuário obtenha informações sobre a função que você acabou de criar. O`iam:PassRole`A permissão permite que o usuário transmita a função para outraAWSserviçoServiço.

```
{
    "Version": "2012-10-17",
    "Statement": {
        "Effect": "Allow",
        "Action": [
            "iam:GetRole",
            "iam:PassRole"
        ],
        "Resource": "arn:aws:iam::your Conta da AWS id:role/your role name"
    }
}
```

3. Crie um alias de função do AWS IoT Core. O dispositivo para o qual fará chamadas diretas paraAWSOs serviços devem saber qual ARN de função usar ao se conectar aoAWS IoT Core. Fixar o ARN da função no código não é uma boa solução porque isso requer que você atualize o dispositivo toda vez que o ARN da função for alterado. Uma solução mais adequada é usar a API `CreateRoleAlias` para criar um alias de função que aponte para o ARN da função. Se o ARN da função for alterado, basta atualizar o alias da função. Nenhuma alteração é necessária no dispositivo. Essa API usa os seguintes parâmetros:

   `roleAlias`

   Obrigatório. Uma string arbitrária que identifica o alias da função. Funciona como chave primária no modelo de dados do alias da função. Contém de 1 a 128 caracteres e deve incluir apenas caracteres alfanuméricos e os símbolos =, @ e -. Caracteres alfabéticos em maiúsculas e minúsculas são permitidos.

   `roleArn`

   Obrigatório. O ARN da função ao qual o alias da função se refere.

   `credentialDurationSeconds`

   Opcional. Por quanto tempo (em segundos) a credencial é válida. O valor mínimo é 900 segundos (15 minutos). O valor máximo é de 43.200 segundos (12 horas). O valor padrão é de 3.600 segundos (1 hora).

   > **Note**
   >
   > OAWS IoT CoreO provedor de credenciais pode emitir uma credencial com um tempo de vida útil máximo de 43.200 segundos (12 horas). Ter a credencial válida por até 12 horas pode ajudar a reduzir o número de chamadas para o provedor de credenciais armazenando em cache a credencial por mais tempo. O`credentialDurationSeconds`O valor deve ser menor ou igual à duração máxima da sessão da função do IAM que o alias de função faz referência ao.

   Para obter mais informações sobre essa API, consulte CreateRoleAlias.

4. Anexe uma política a um certificado de dispositivo. A política anexada ao certificado do dispositivo deve conceder permissão ao dispositivo para assumir a função. Para isso, conceda permissão para a ação `iot:AssumeRoleWithCertificate` do alias da função, como no exemplo a seguir.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":"iot:AssumeRoleWithCertificate",
         "Resource":"arn:aws:iot:your region:your_aws_account_id:rolealias/your role
 alias"
      }
```

```
    ]
}
```

5. Faça uma solicitação HTTPS ao provedor de credenciais para receber um token de segurança. Forneça as informações a seguir:

   - Certificado: Como se trata de uma solicitação HTTP por meio de autenticação TLS mútua, é necessário fornecer o certificado e a chave privada para o cliente quando estiver fazendo a solicitação. Use o certificado e a chave privada que você usou quando registrou o certificado no AWS IoT Core.

     Para garantir que o dispositivo esteja se comunicando com o AWS IoT Core (e não um serviço se passando por ele), consulte Autenticação de servidor, siga os links para fazer download dos certificados CA apropriados e copie-os no dispositivo.
   - RoleAlias: O nome do alias de função que você criou para o provedor de credenciais.
   - ThingName: O nome que você criou ao registrar oAWS IoT Corecoisa do Ele é passado como o valor do cabeçalho HTTP `x-amzn-iot-thingname`. Esse valor é necessário somente quando você usa atributos de coisas como variáveis de política nas políticas da AWS IoT Core ou do IAM.

     > **Note**
     >
     > OThingNameque você fornece`x-amzn-iot-thingname`Deve corresponder ao nome doAWS IoTRecurso de coisa atribuído a um certificado. Se não corresponder, um erro 403 será retornado.

   Execute o comando a seguir naAWS CLIPara obter o endpoint do provedor de credenciais doConta da AWS. Para obter mais informações sobre essa API, consulte DescribeEndpoint.

```
aws iot describe-endpoint --endpoint-type iot:CredentialProvider
```

   O objeto JSON a seguir é uma saída de exemplo do comando describe-endpoint. Ele contém o `endpointAddress` que você usa para solicitar um token de segurança.

```
{
    "endpointAddress": "your_aws_account_specific_prefix.credentials.iot.your
 region.amazonaws.com"
}
```

   Use o endpoint para fazer uma solicitação HTTPS ao provedor de credenciais para que retorne um token de segurança. O comando de exemplo a seguir usa `curl`, mas é possível usar qualquer cliente HTTP.

```
curl --cert your certificate --key your device certificate key pair -H "x-amzn-iot-
thingname: your thing name" --cacert AmazonRootCA1.pem https://your endpoint /role-
aliases/your role alias/credentials
```

   Esse comando retorna um objeto de token de segurança que contém um `accessKeyId`, uma `secretAccessKey`, um `sessionToken`e uma expiração. O objeto JSON a seguir é uma saída de exemplo do comando `curl`.

```
    {"credentials":{"accessKeyId":"access key","secretAccessKey":"secret access
 key","sessionToken":"session token","expiration":"2018-01-18T09:18:06Z"}}
```

Você pode usar o`accessKeyId`,`secretAccessKey`, e`sessionToken`valores para assinar solicitações paraAWSServiços da . Para um end-to-end demonstração, vejaComo eliminar a necessidade de codificação rígidaAWSCredenciais em dispositivos usando oAWS IoTProvedor de credenciaisPublicação no blog noAWSBlog de segurança.

Acesso entre contas com o IAM

AWS IoT CoreO permite habilitar um principal para publicar ou assinar um tópico definido em umConta da AWSNão pertencente ao diretor. Você configura acesso entre contas criando uma política do IAM e uma função do IAM e, em seguida, anexando a política à função.

Primeiro, crie uma política do IAM gerenciada pelo cliente, conforme descrito emCriação de políticas do IAM, assim como você faria com outros usuários e certificados em seuConta da AWS.

Para dispositivos registrados emAWS IoT CoreRegistrar, a política a seguir concede permissão aos dispositivos se conectam aoAWS IoT Coreusando um ID de cliente que corresponda ao nome da coisa do dispositivo e para publicar no`my/topic/`*`thing-name`* onde*`Nome da coisa`*é o nome do dispositivo:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:Connect"
            ],
            "Resource": ["arn:aws:iot:us-east-1:123456789012:client/
${iot:Connection.Thing.ThingName}"]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iot:Publish"
            ],
            "Resource": ["arn:aws:iot:us-east-1:123456789012:topic/my/topic/
${iot:Connection.Thing.ThingName}"],
        }
    ]
}
```

Para dispositivos não registrados no registro do AWS IoT Core, a política a seguir concede permissão a um dispositivo para usar o nome da coisa `client1` registrado no registro do AWS IoT Core da conta (123456789012) para se conectar ao AWS IoT Core e para publicar em um tópico específico do ID do cliente cujo nome é prefixado com `my/topic/`:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:Connect"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:client/client1"
            ]
        },
        {
            "Effect": "Allow",
```

```
            "Action": [
                "iot:Publish"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:topic/my/topic/${iot:ClientId}"
            ]
        }
    ]
}
```

Em seguida, siga as etapas em Criação de uma função para conceder permissões a um usuário do IAM. Insira o ID da conta doConta da AWSCom o qual você deseja compartilhar o acesso. Em seguida, na etapa final, anexe a política recém-criada à função. Se, posteriormente, você precisar modificar aAWSID de conta ao qual você está concedendo acesso, você pode usar o formato de política de confiança a seguir para fazer isso:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam:us-east-1:567890123456:user:MyUser"
            },
            "Action": "sts:AssumeRole",
        }
    ]
}
```

Proteção de dados no AWS IoT Core

O modelo de responsabilidade compartilhada da AWS se aplica à proteção de dados no AWS IoT Core. Conforme descrito nesse modelo, a AWS é responsável por proteger a infraestrutura global que executa toda a Nuvem AWS. Você é responsável por manter o controle sobre seu conteúdo hospedado nessa infraestrutura. Esse conteúdo inclui as tarefas de configuração e gerenciamento de segurança dos Serviços da AWS que você usa. Para obter mais informações sobre a privacidade de dados, consulte as Perguntas frequentes sobre privacidade de dados. Para obter mais informações sobre a proteção de dados na Europa, consulte a postagem do blog AWS Shared Responsibility Model and GDPR no Blog de segurança da AWS.

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

- Use uma autenticação multifator (MFA) com cada conta.
- Use SSL/TLS para se comunicar com os recursos da AWS. Recomendamos TLS 1.2 ou posterior.
- Configure o registro em log das atividades da API e do usuário com o AWS CloudTrail.
- Use as soluções de criptografia da AWS, juntamente com todos os controles de segurança padrão nos serviços da AWS.
- Use serviços gerenciados de segurança avançada, como o Amazon Macie, que ajuda a localizar e proteger dados pessoais armazenados no Amazon S3.
- Se você precisar de módulos criptográficos validados pelo FIPS 140-2 ao acessar a AWS por meio de uma interface de linha de comando ou uma API, use um endpoint do FIPS. Para obter mais informações sobre endpoints do FIPS, consulte Federal Information Processing Standard (FIPS) 140-2.

É altamente recomendável que você nunca coloque informações de identificação confidenciais, como endereços de e-mail dos seus clientes, em marcações ou campos de formato livre, como um campo Name (Nome). Isso também vale para o uso do AWS IoT ou de outros serviços da AWS com o console, a API, a AWS CLI ou os AWS SDKs. Quaisquer dados inseridos em marcações ou campos de formato livre usados para nomes podem ser usados para logs de cobrança ou diagnóstico. Se você fornecer um URL para um servidor externo, recomendemos fortemente que não sejam incluídas informações de credenciais no URL para validar a solicitação a esse servidor.

Para obter mais informações sobre proteção de dados, consulte a publicação Modelo de responsabilidade compartilhada da AWS e do GDPR no Blog de segurança da AWS.

Os dispositivos do AWS IoT coletam dados, realizam uma manipulação desses dados e enviam esses dados para outro web service. É possível optar por armazenar alguns dados em seu dispositivo por um curto período. Você é responsável por fornecer qualquer proteção de dados sobre esses dados em repouso. Quando o dispositivo envia dados paraAWS IoT, ele o faz por meio de uma conexão TLS, conforme discutido mais adiante nesta seção.AWS IoTdispositivos podem enviar dados para qualquerAWSserviçoServiço. Para obter mais informações sobre a segurança de dados de cada serviço, consulte a documentação desse serviço.AWS IoTPode ser configurado para gravar logs em CloudWatch Logs e logAWS IoTChamadas de API para oAWS CloudTrail. Para obter mais informações sobre segurança de dados desses serviços, consulteAutenticação e controle de acesso para Amazon CloudWatcheCriptografia CloudTrail Registrar arquivos com oAWSChaves gerenciadas pelo KMS.

Criptografia de dados no AWS IoT

Por padrão, todosAWS IoTOs dados em trânsito e em repouso são criptografados. Os dados em trânsito são criptografados com TLS (p. 381)e os dados em repouso são criptografados usandoAWSchaves de propriedade.AWS IoTNo momento, o não oferece suporte à gerenciada pelo clienteAWS KMS keys(chaves KMS) deAWSKey Management Service (AWS KMS); no entanto, Device Advisor eAWS IoTUso sem fio apenas umChave pertencente à AWSpara criptografar dados do cliente.

Segurança de transporte no AWS IoT

OAWS IoTO agente de mensagens da e o serviço Device Shadow criptografam todas as comunicações durante o trânsito usandoTLS versão 1.2. O TLS é usado para garantir a confidencialidade dos protocolos de aplicativos (MQTT, HTTP e WebSocket) com suporte pelaAWS IoT. O suporte ao TLS está disponível em várias de linguagens de programação e sistemas operacionais. Dados dentroAWSé criptografado pelo específicoAWSserviçoServiço. Para obter mais informações sobre criptografia de dados em outros serviços da AWS, consulte a documentação de segurança desse serviço.

Para MQTT, o TLS criptografa a conexão entre o dispositivo e o operador. A autenticação do cliente TLS é usada pela AWS IoT para identificar os dispositivos. Para HTTP, o TLS criptografa a conexão entre o dispositivo e o operador. A autenticação é delegada aAWSSignature versão 4.

A AWS IoT requer que os dispositivos enviem a extensão SNI (Server Name Indication) para o protocolo TLS (Transport Layer Security) e forneçam o endereço completo do endpoint no campo `host_name`. O campo `host_name` deve conter o endpoint que você está chamando e deve ser:

- O `endpointAddress` retornado por `aws iot describe-endpoint --endpoint-type iot:Data-ATS`

  ou
- O `domainName` retornado por `aws iot describe-domain-configuration --domain-configuration-name "domain_configuration_name"`

Conexões tentadas por dispositivos sem o correto`host_name`o valor falhará eAWS IoTregistrará falhas para CloudWatch se o tipo de autenticação for Autenticação personalizada.

AWS IoTO não oferece suporte ao SessionTicket Extensão TLS.

Segurança de transporte para dispositivos sem fio LoRaWAN

Os dispositivos LoraWAN seguem as práticas de segurança descritas emSEGURANÇA LOraWAN™: Um white paper preparado para o LoRa Alliance™ de Gemalto, Actility e Semtech.

Para obter mais informações sobre segurança de transporte com dispositivos LoRawan, consulteSegurança de dados com oAWS IoT Corepara LoRaWAN (p. 1217).

Suporte ao pacote de criptografia do TLS

A AWS IoT oferece suporte aos seguintes pacotes de criptografia:

- ECDHE-ECDSA-AES128-GCM-SHA256 (recomendado)
- ECDHE-RSA-AES128-GCM-SHA256 (recomendado)
- ECDHE-ECDSA-AES128-SHA256
- ECDHE-RSA-AES128-SHA256
- ECDHE-ECDSA-AES128-SHA
- ECDHE-RSA-AES128-SHA
- ECDHE-ECDSA-AES256-GCM-SHA384
- ECDHE-RSA-AES256-GCM-SHA384
- ECDHE-ECDSA-AES256-SHA384
- ECDHE-RSA-AES256-SHA384
- ECDHE-RSA-AES256-SHA
- ECDHE-ECDSA-AES256-SHA
- AES128-GCM-SHA256
- AES128-SHA256
- AES128-SHA
- AES256-GCM-SHA384
- AES256-SHA256
- AES256-SHA

Criptografia de dados no AWS IoT

A proteção de dados protege os dados em trânsito (à medida que são transferidos para e do AWS IoT) e em repouso (enquanto estão armazenados em dispositivos ou por outros serviços da AWS). Todos os dados enviados aoAWS IoTé enviado por uma conexão TLS usando MQTT, HTTPS e WebSocket protocolos, tornando-o seguro por padrão durante o trânsito.AWS IoTos dispositivos coletam dados e os enviam para outrosAWSserviços para processamento posterior. Para obter mais informações sobre criptografia de dados em outros serviços da AWS, consulte a documentação de segurança desse serviço.

O FreeRTOS fornece uma biblioteca PKCS #11 que abstrai o armazenamento de chaves, acessando objetos criptográficos e gerenciando sessões. É sua responsabilidade usar essa biblioteca para criptografar dados em repouso em seus dispositivos. Para obter mais informações, consulteBiblioteca do padrão de criptografia de chave pública FreeRTOS (PKCS) #11.

Device Advisor

Criptografia em trânsito

Os dados enviados de e para o Device Advisor são criptografados em trânsito. Todos os dados enviados de e para o serviço ao usar as APIs do Device Advisor são criptografados usando o Signature Versão 4. Para obter mais informações sobre comoAWSAs solicitações de API são assinadas, consulteSigningAWSSolicitações da API do. Todos os dados enviados de seus dispositivos de teste para o endpoint de teste do Device Advisor são enviados por meio de uma conexão TLS para que fiquem seguros por padrão em trânsito.

Gerenciamento de chaves no AWS IoT

Todas as conexões ao AWS IoT são feitas usando TLS, portanto, nenhuma chave de criptografia no lado do cliente é necessária para a conexão TLS inicial.

Os dispositivos devem autenticar usando um certificado X.509 ou um Amazon Cognito Identity. É possível fazer com que o AWS IoT gere um certificado para você, e, no caso, ele gerará um par de chaves públicas/privadas. Se você estiver usando o console do AWS IoT, será solicitado a fazer download do certificado e das chaves. Se você estiver usando o`create-keys-and-certificate`O comando da CLI, o certificado e as chaves são retornados pelo comando da CLI. Você é responsável por copiar o certificado e a chave privada em seu dispositivo e mantê-lo seguro.

AWS IoTNo momento, o não oferece suporte à gerenciada pelo clienteAWS KMS keys(chaves KMS) deAWS Key Management Service(AWS KMS); no entanto, Device Advisor eAWS IoTUso sem fio apenas umChave pertencente à AWSpara criptografar dados do cliente.

Device Advisor

Todos os dados enviados ao Device Advisor ao usar oAWSAs APIs são criptografadas em repouso. O Device Advisor criptografa todos os seus dados em repouso usando chaves KMS armazenadas e gerenciadas emAWS Key Management Service. O Device Advisor criptografa seus dados usando oChaves pertencentes à AWS. Para obter mais informações sobre o Chaves pertencentes à AWS, consulte Chaves pertencentes à AWS.

Identity and Access Management para o AWS IoT

O AWS Identity and Access Management (IAM) é um serviço da AWS service (Serviço da AWS) que ajuda a controlar o acesso aos recursos da AWS de forma segura. Os administradores do IAM controlam quem pode ser autenticado (conectado) e autorizado (ter permissões) a usar os recursos do AWS IoT. O IAM é um AWS service (Serviço da AWS) que pode ser usado sem custo adicional.

Tópicos

- Público (p. 384)
- Autenticar com identidades do IAM (p. 384)
- Gerenciamento do acesso usando políticas (p. 386)
- Como o AWS IoT funciona com o IAM (p. 388)
- Exemplos de políticas baseadas em identidade do AWS IoT (p. 404)
- Políticas gerenciadas pela AWS para o AWS IoT (p. 407)
- Solução de problemas de identidade e acesso do AWS IoT (p. 417)

Público

O uso do AWS Identity and Access Management (IAM) varia dependendo do trabalho que for realizado no AWS IoT.

Usuário do serviço – Se você usar o serviço AWS IoT para fazer o trabalho, o administrador fornecerá as credenciais e as permissões necessárias. À medida que usar mais recursos do AWS IoT para fazer seu trabalho, você poderá precisar de permissões adicionais. Entender como o acesso é gerenciado pode ajudar você a solicitar as permissões corretas ao seu administrador. Se não for possível acessar um recurso no AWS IoT, consulte Solução de problemas de identidade e acesso do AWS IoT (p. 417).

Administrador do serviço – Se você for o responsável pelos recursos do AWS IoT na empresa, provavelmente terá acesso total ao AWS IoT. Seu trabalho é determinar quais recursos do AWS IoT seus funcionários devem acessar. Assim, é necessário enviar solicitações ao administrador do IAM para alterar as permissões dos usuários de seu serviço. Revise as informações nesta página para entender os conceitos básicos do IAM. Para saber mais sobre como a empresa pode usar o IAM com o AWS IoT, consulte Como o AWS IoT funciona com o IAM (p. 388).

Administrador do IAM – Se você for um administrador do IAM, talvez queira saber detalhes sobre como pode gravar políticas para gerenciar o acesso ao AWS IoT. Para visualizar exemplos de políticas baseadas em identidade do AWS IoT que podem ser usadas no IAM, consulte Exemplos de políticas baseadas em identidade do AWS IoT (p. 404).

Autenticar com identidades do IAM

DentroAWS IoTAs identidades do podem ser certificados de dispositivo (X.509), identidades do Amazon Cognito ou usuários ou grupos do IAM. Este tópico aborda apenas identidades do IAM. Para obter mais informações sobre as outras identidades aos quais o AWS IoT oferece suporte, consulte Autenticação de cliente (p. 297).

A autenticação é a forma como você faz login na AWS usando suas credenciais de identidade. Para obter mais informações sobre como fazer login usando o AWS Management Console, consulte Login no AWS Management Console como usuário do IAM ou usuário root no Guia do usuário do IAM.

É necessário estar autenticado (conectado à AWS) como o usuário root da Conta da AWS ou um usuário do IAM, ou ainda assumindo uma função do IAM. Também é possível usar a autenticação de logon único da sua empresa ou até mesmo fazer login usando o Google ou o Facebook. Nesses casos, o administrador configurou anteriormente federação de identidades usando funções do IAM. Ao acessar a AWS usando credenciais de outra empresa, você estará assumindo uma função indiretamente.

Para fazer login diretamente no AWS Management Console, use sua senha com o e-mail do usuário root ou seu nome de usuário do IAM. É possível acessar a AWS de maneira programática usando chaves de acesso do seu usuário root ou dos usuários do IAM. AWS fornece ferramentas SDK e de linha de comando para assinar de forma criptográfica a sua solicitação usando suas credenciais. Se você não utilizar as ferramentas AWS, você deverá assinar a solicitação por conta própria. Faça isso usando o Signature versão 4, um protocolo para autenticação de solicitações de API de entrada. Para obter mais informações sobre solicitações de autenticação, consulte Processo de assinatura do Signature Version 4 na Referência geral da AWS.

Independentemente do método de autenticação usado, também pode ser exigido que você forneça informações adicionais de segurança. Por exemplo, a AWS recomenda o uso da autenticação multifator (MFA) para aumentar a segurança de sua conta. Para saber mais, consulte Uso da autenticação multifator (MFA) AWS no Guia do usuário do IAM.

Usuário root da Conta da AWS

Ao criar uma Conta da AWS, você começa com uma única identidade de login que tenha acesso total a todos os recursos e Serviços da AWS na conta. Essa identidade, denominada Conta da AWSusuário

raiz da , é acessada por login com o endereço de e-mail e a senha que você usou para criar a conta. É recomendável não usar o usuário raiz para suas tarefas diárias, nem mesmo para as administrativas. Em vez disso, siga as práticas recomendadas sobre utilização de usuário raiz somente para criar seu primeiro usuário do IAM. Depois, armazene as credenciais do usuário raiz com segurança e use-as para executar somente algumas tarefas de gerenciamento de contas e de serviços.

Grupos e usuários do IAM

Um usuário do IAM é uma identidade dentro da Conta da AWS que tem permissões específicas para uma única pessoa ou aplicação. Um usuário do IAM pode ter credenciais de longo prazo, como um nome de usuário e uma senha ou um conjunto de chaves de acesso. Para saber como gerar chaves de acesso, consulte Gerenciar chaves de acesso para usuários do IAM no Guia do usuário do IAM. Ao gerar chaves de acesso para um usuário do IAM, visualize e salve o par de chaves de maneira segura. Não será possível recuperar a chave de acesso secreta futuramente. Em vez disso, você deverá gerar outro par de chaves de acesso.

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

Usuários são diferentes de funções. Um usuário é exclusivamente associado a uma pessoa ou a uma aplicação, mas uma função pode ser assumida por qualquer pessoa que precisar dela. Os usuários têm credenciais permanentes de longo prazo, mas as funções fornecem credenciais temporárias. Para saber mais, consulte Quando criar um usuário do IAM (em vez de uma função) no Guia do usuário do IAM.

Funções do IAM

Uma função do IAM é uma identidade dentro da Conta da AWS que tem permissões específicas. Ela é semelhante a um usuário do IAM, mas não está associada a uma pessoa específica. É possível assumir temporariamente um perfil do IAM no AWS Management Console alternando funções. É possível assumir um perfil chamando uma operação de API da AWS CLI ou da AWS, ou usando um URL personalizado. Para mais informações sobre métodos para o uso de perfis, consulte Usar perfis do IAM no Guia do usuário do IAM.

Os perfis do IAM com credenciais temporárias são úteis nas seguintes situações:

- Permissões temporárias para usuários do IAM: um usuário do IAM pode assumir um perfil do IAM para obter temporariamente permissões diferentes para uma tarefa específica.
- Acesso de usuário federado: em vez de criar um usuário do IAM, você poderá usar identidades de usuários existentes no AWS Directory Service, em seu diretório de usuários corporativos ou em um provedor de identidades da Web. Eles são conhecidos como usuários federados. A AWS atribui uma função a um usuário federado quando o acesso é solicitado por meio de um provedor de identidades. Para obter mais informações sobre usuários federados, consulte Usuários federados e perfis no Guia do usuário do IAM.
- Acesso entre contas: é possível usar um perfil do IAM para permitir que alguém (um principal confiável) em outra conta acesse recursos em sua conta. As funções são a principal forma de conceder acesso entre contas. No entanto, alguns Serviços da AWS permitem que você anexe uma política diretamente a um recurso (em vez de usar uma função como proxy). Para saber 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 aplicações no Amazon EC2 ou armazene objetos no Amazon S3. Um serviço pode fazer isso usando as permissões do principal de chamada, usando uma função de serviço ou uma função vinculada ao serviço.

  - Permissões de principal: ao usar um usuário ou uma função do IAM para executar ações na AWS, você é considerado um principal. As políticas concedem permissões a uma entidade principal. Quando você usa alguns serviços, pode executar uma ação que, em seguida, aciona outra ação em outro serviço. Nesse caso, você deve ter permissões para executar ambas as ações. Para ver se uma ação exige ações dependentes adicionais em uma política, consulte Ações, recursos e chaves de condição para o AWS IoT na Referência de autorização do serviço.
  - 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 Criar uma função para delegar permissões a um AWS service (Serviço da AWS) no Guia do usuário do IAM.
  - Função vinculada a serviço: uma função vinculada a serviço é um tipo de função de serviço vinculada a um AWS service (Serviço da AWS). O serviço pode assumir a função de executar uma ação em seu nome. As funções vinculadas ao serviço aparecem em sua conta do IAM e são de propriedade do serviço. Um administrador do IAM pode visualizar, mas não pode editar as permissões para funções vinculadas ao serviço.
- Aplicações em execução no Amazon EC2: é possível usar uma função do IAM para gerenciar credenciais temporárias para aplicações em execução em uma instância do EC2 e fazer solicitações da AWS CLI ou da AWS API. É preferível fazer isso a armazenar chaves de acesso na instância do EC2. Para atribuir uma função da AWS a uma instância do EC2 e disponibilizá-la para todas as suas aplicações, crie um perfil de instância que esteja anexado a ela. Um perfil de instância contém a função 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 aplicações em execução nas instâncias do Amazon EC2 no Guia do usuário do IAM.

Para saber se deseja usar as funções do IAM, consulte Quando criar uma função do IAM (em vez de um usuário) no Guia do usuário do IAM.

Gerenciamento do acesso usando políticas

Você controla o acesso na AWS criando e anexando políticas às identidades do IAM ou aos recursos da AWS. Uma política é um objeto na AWS que, quando associado a uma identidade ou recurso, define suas permissões. Você pode fazer login como o usuário raiz ou um usuário do IAM ou assumir uma função do IAM. Quando você faz uma solicitação, a AWS avalia as políticas relacionadas baseadas em identidade ou baseadas em recursos. As permissões nas políticas determinam se a solicitação será permitida ou negada. A maioria das políticas são armazenadas na 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 as políticas JSON da AWS para especificar quem tem acesso a quê. Ou seja, qual principal pode executar ações em quais recursos, e em que condições.

Cada entidade do IAM (usuário ou função) começa sem permissões. Em outras palavras, por padrão, os usuários não podem fazer nada, nem mesmo alterar sua própria senha. Para dar permissão a um usuário para fazer algo, um administrador deve anexar uma política de permissões ao usuário. Ou o administrador pode adicionar o usuário a um grupo que tenha as permissões pretendidas. Quando um administrador concede permissões a um grupo, todos os usuários desse grupo recebem essas permissões.

As políticas do IAM definem permissões para uma ação, independentemente do método usado para executar a operação. Por exemplo, suponha que você tenha uma política que permite a ação `iam:GetRole`. Um usuário com essa política pode obter informações de funções do AWS Management Console, da AWS CLI ou da API da AWS.

Políticas baseadas em identidade

As políticas baseadas em identidade são documentos de políticas de permissões JSON que você pode anexar a uma identidade, como usuário, grupo de usuários ou função do IAM. Essas políticas controlam

quais ações os usuários e funções podem realizar, em quais recursos e em que 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 podem ser categorizadas ainda mais como políticas em linha ou políticas gerenciadas. As políticas em linha são anexadas diretamente a um único usuário, grupo ou função. As políticas gerenciadas são políticas independentes que podem ser anexadas a vários usuários, grupos e funções na Conta da AWS. As políticas gerenciadas incluem políticas gerenciadas pela AWS e políticas gerenciadas pelo cliente. Para saber como escolher entre uma política gerenciada ou uma política em linha, consulte Escolher entre políticas gerenciadas e políticas em linha no Guia do usuário do IAM.

Políticas baseadas em recursos

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

Políticas baseadas em recursos são políticas em linha que estão localizadas nesse serviço. Não é possível usar as políticas gerenciadas da AWS do IAM em uma política baseada em recursos.

Listas de controle de acesso (ACLs)

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

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

Outros tipos de política

A AWS oferece suporte a tipos de política 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 da entidade e seus limites de permissões. As políticas baseadas em recurso que especificam o usuário ou a função no campo `Principal` não são limitadas pelo limite de permissões. Uma negação explícita em qualquer uma dessas políticas substitui a permissão. Para obter mais informações sobre limites de permissões, consulte Limites de permissões para identidades do IAM no Guia do usuário do IAM.
- Políticas de controle de serviço (SCPs): SCPs são políticas JSON que especificam as permissões máximas para uma organização ou unidade organizacional (UO) no AWS Organizations. O AWS Organizations é um serviço para agrupar e gerenciar centralmente várias Contas da AWS pertencentes à sua empresa. Se você habilitar todos os recursos em uma organização, poderá aplicar políticas de controle de serviço (SCPs) a qualquer uma ou a todas as contas. Uma SCP limita as permissões para entidades em contas-membro, incluindo cada usuário root da Conta da AWS. Para obter mais informações sobre o Organizations e SCPs, consulte Como os SCPs funcionam no Guia do usuário do AWS Organizations.
- 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 da função e

das políticas de sessão. As permissões também podem ser provenientes de uma política baseada em recurso. 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 a AWS determina se deve permitir uma solicitação quando há vários tipos de política envolvidos, consulte Lógica da avaliação de políticas no Guia do usuário do IAM.

Como o AWS IoT funciona com o IAM

Antes de usar o IAM para gerenciar o acesso aoAWS IoT, você deve entender quais recursos do IAM estão disponíveis para uso comAWS IoT. Para ter uma visão geral de como o AWS IoT e outros serviços da AWS funcionam com o IAM, consulte Serviços da AWS compatíveis com o IAM no Guia do usuário do IAM.

Tópicos

- Políticas baseadas em identidade do AWS IoT (p. 388)
- Políticas baseadas em recursos do AWS IoT (p. 404)
- Autorização baseada em tags do AWS IoT (p. 404)
- Funções do IAM no AWS IoT (p. 404)

Políticas baseadas em identidade do AWS IoT

Com as políticas baseadas em identidade do API Gateway, é possível especificar ações ou recursos permitidos ou negados, bem como as condições sob as quais as ações são permitidas ou negadas.AWS IoTO oferece suporte a ações, recursos e chaves de condição específicos. Para conhecer todos os elementos usados em uma política JSON, consulte Referência de elementos de política JSON do IAM no Manual do usuário do IAM.

Ações

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

O elemento `Action` de uma política JSON descreve as ações que você pode usar para permitir ou negar acesso em uma política. As ações de política geralmente têm o mesmo nome que a operação de API da AWS associada. Existem algumas exceções, como ações somente de permissão, que não têm uma operação de API correspondente. Há também algumas operações que exigem várias ações em uma política. Essas ações adicionais são chamadas de ações dependentes.

Inclua ações em uma política para conceder permissões para executar a operação associada.

A tabela a seguir lista as ações do IAM IoT, as associadasAWS IoTAPI e o recurso que a ação manipula.

Ações de políticas	AWS IoT API	Recursos
iot:AcceptCertificateTransfer	AcceptCertificateTransfer	`arn:aws:iot:region:account-id:cert/cert-id` Note OConta da AWSEspecificada no ARN deve ser a conta para a qual o certificado está sendo transferido.

Ações de políticas	AWS IoT API	Recursos

iot:AddThingToThingGroup	AddThingToThingGroup	`arn:aws:iot:`*`region`*`:`*`account-id`*`:thinggroup/`*`thing-group-name`
		
		`arn:aws:iot:`*`region`*`:`*`account-id`*`:thing/`*`thing-name`

iot:AssociateTargetsWithJob	AssociateTargetsWithJob	Job

iot:AttachPolicy	AttachPolicy	`arn:aws:iot:`*`region`*`:`*`account-id`*`:thinggroup/`*`thing-group-name`
		
		ou
		
		`arn:aws:iot:`*`region`*`:`*`account-id`*`:cert/`*`cert-id`

iot:AttachPrincipalPolicy	AttachPrincipalPolicy	`arn:aws:iot:`*`region`*`:`*`account-id`*`:cert/`*`cert-id`

iot:AttachSecurityProfile	AttachSecurityProfile	`arn:aws:iot:`*`region`*`:`*`account-id`*`:securityprofile/`*`security-profile-name`
		
		`arn:aws:iot:`*`region`*`:`*`account-id`*`:dimension/`*`dimension-name`

iot:AttachThingPrincipal	AttachThingPrincipal	`arn:aws:iot:`*`region`*`:`*`account-id`*`:cert/`*`cert-id`
iot:CancelCertificateTransfer	CancelCertificateTransfer	`arn:aws:iot:`*`region`*`:`*`account-id`*`:cert/`*`cert-id` **Note** OConta da AWSEspecificada no ARN deve ser a conta para a qual o certificado está sendo transferido.
iot:CancelJob	CancelJob	`arn:aws:iot:`*`region`*`:`*`account-id`*`:job/`*`job-id`

iot:CancelJobExecution	CancelJobExecution	`arn:aws:iot:`*`region`*`:`*`account-id`*`:job/`*`job-id`
		
		`arn:aws:iot:`*`region`*`:`*`account-id`*`:thing/`*`thing-name`

iot:ClearDefaultAuthorizer	ClearDefaultAuthorizer	Nenhum
iot:CreateAuthorizer	CreateAuthorizer	`arn:aws:iot:`*`region`*`:`*`account-id`*`:authorizer/`*`authorizer-function-name`
iot:CreateCertificateFromCsr	CreateCertificateFromCsr	*
iot:CreateDimension	CreateDimension	`arn:aws:iot:`*`region`*`:`*`account-id`*`:dimension/`*`dimension-name`

iot:CreateJob	CreateJob	`arn:aws:iot:`*`region`*`:`*`account-id`*`:job/`*`job-id`
		
		`arn:aws:iot:`*`region`*`:`*`account-id`*`:thinggroup/`*`thing-group-name`
		
		`arn:aws:iot:`*`region`*`:`*`account-id`*`:thing/`*`thing-name`
		
		`arn:aws:iot:`*`region`*`:`*`account-id`*`:jobtemplate/`*`job-template-id`

Ações de políticas	AWS IoT API	Recursos

IOT: CreateJOBTemplate	CreateJobTemplate	`arn:aws:iot:region:account-id:job/job-id`
		
		`arn:aws:iot:region:account-id:jobtemplate/job-template-id`

iot:CreateKeysAndCertificate	CreateKeysAndCertificate	*
iot:CreatePolicy	CreatePolicy	`arn:aws:iot:region:account-id:policy/policy-name`
iot:CreatePolicyVersion	CreatePolicyVersion	`arn:aws:iot:region:account-id:policy/policy-name` Note Isso deve ser umAWS IoTpolítica, não uma política do IAM.
iot:CreateRoleAlias	CreateRoleAlias	(parâmetro: roleAlias) `arn:aws:iot:region:account-id:rolealias/role-alias-name`

iot:CreateSecurityProfile	CreateSecurityProfile	`arn:aws:iot:region:account-id:securityprofile/security-profile-name`
		
		`arn:aws:iot:region:account-id:dimension/dimension-name`

iot:CreateThing	CreateThing	`arn:aws:iot:region:account-id:thing/thing-name`
iot:CreateThingGroup	CreateThingGroup	`arn:aws:iot:region:account-id:thinggroup/thing-group-name` para o grupo que está sendo criado e para o grupo pai, se usado
iot:CreateThingType	CreateThingType	`arn:aws:iot:region:account-id:thingtype/thing-type-name`
iot:CreateTopicRule	CreateTopicRule	`arn:aws:iot:region:account-id:rule/rule-name`
iot:DeleteAuthorizer	DeleteAuthorizer	`arn:aws:iot:region:account-id:authorizer/authorizer-name`
iot:DeleteCACertificate	DeleteCACertificate	`arn:aws:iot:region:account-id:cacert/cert-id`
iot:DeleteCertificate	DeleteCertificate	`arn:aws:iot:region:account-id:cert/cert-id`
iot:DeleteDimension	DeleteDimension	`arn:aws:iot:region:account-id:dimension/dimension-name`
iot:DeleteJob	DeleteJob	`arn:aws:iot:region:account-id:job/job-id`
IOT: DeleteJobTemplate	DeleteJobTemplate	`arn:aws:iot:region:account-id:job/job-template-id`

Ações de políticas	AWS IoT API	Recursos

iot:DeleteJobExecution	DeleteJobExecution	`arn:aws:iot:region:account-id:job/job-id`
		
		`arn:aws:iot:region:account-id:thing/thing-name`

iot:DeletePolicy	DeletePolicy	`arn:aws:iot:region:account-id:policy/policy-name`
iot:DeletePolicyVersion	DeletePolicyVersion	`arn:aws:iot:region:account-id:policy/policy-name`
iot:DeleteRegistrationCode	DeleteRegistrationCode	*
iot:DeleteRoleAlias	DeleteRoleAlias	`arn:aws:iot:region:account-id:rolealias/role-alias-name`

iot:DeleteSecurityProfile	DeleteSecurityProfile	`arn:aws:iot:region:account-id:securityprofile/security-profile-name`
		
		`arn:aws:iot:region:account-id:dimension/dimension-name`

iot:DeleteThing	DeleteThing	`arn:aws:iot:region:account-id:thing/thing-name`
iot:DeleteThingGroup	DeleteThingGroup	`arn:aws:iot:region:account-id:thinggroup/thing-group-name`
iot:DeleteThingType	DeleteThingType	`arn:aws:iot:region:account-id:thingtype/thing-type-name`
iot:DeleteTopicRule	DeleteTopicRule	`arn:aws:iot:region:account-id:rule/rule-name`
iot:DeleteV2LoggingLevel	DeleteV2LoggingLevel	`arn:aws:iot:region:account-id:thinggroup/thing-group-name`
iot:DeprecateThingType	DeprecateThingType	`arn:aws:iot:region:account-id:thingtype/thing-type-name`
iot:DescribeAuthorizer	DescribeAuthorizer	`arn:aws:iot:region:account-id:authorizer/authorizer-function-name` (parâmetro: authorizerName) none
iot:DescribeCACertificate	DescribeCACertificate	`arn:aws:iot:region:account-id:cacert/cert-id`
iot:DescribeCertificate	DescribeCertificate	`arn:aws:iot:region:account-id:cert/cert-id`
iot:DescribeDefaultAuthorizer	DescribeDefaultAuthorizer	Nenhum
iot:DescribeEndpoint	DescribeEndpoint	*
iot:DescribeEventConfigurations	DescribeEventConfigurations	*
iot:DescribeIndex	DescribeIndex	`arn:aws:iot:region:account-id:index/index-name`
iot:DescribeJob	DescribeJob	`arn:aws:iot:region:account-id:job/job-id`
iot:DescribeJobExecution	DescribeJobExecution	Nenhum

Ações de políticas	AWS IoT API	Recursos
IOT:DescribeJobTemplate	DescribeJobTemplate	arn:aws:iot:*region*:*account-id*:job/*job-template-id*
iot:DescribeRoleAlias	DescribeRoleAlias	arn:aws:iot:*region*:*account-id*:rolealias/*role-alias-name*
iot:DescribeThing	DescribeThing	arn:aws:iot:*region*:*account-id*:thing/*thing-name*
iot:DescribeThingGroup	DescribeThingGroup	arn:aws:iot:*region*:*account-id*:thinggroup/*thing-group-name*
iot:DescribeThingRegistrationTask	DescribeThingRegistrationTask	Nenhum
iot:DescribeThingType	DescribeThingType	arn:aws:iot:*region*:*account-id*:thingtype/*thing-type-name*
iot:DetachPolicy	DetachPolicy	arn:aws:iot:*region*:*account-id*:cert/*cert-id* ou arn:aws:iot:*region*:*account-id*:thinggroup/*thing-group-name*
iot:DetachPrincipalPolicy	DetachPrincipalPolicy	arn:aws:iot:*region*:*account-id*:cert/*cert-id*
iot:DetachSecurityProfile	DetachSecurityProfile	arn:aws:iot:*region*:*account-id*:securityprofile/*security-profile-name* arn:aws:iot:*region*:*account-id*:dimension/*dimension-name*
iot:DetachThingPrincipal	DetachThingPrincipal	arn:aws:iot:*region*:*account-id*:cert/*cert-id*
iot:DisableTopicRule	DisableTopicRule	arn:aws:iot:*region*:*account-id*:rule/*rule-name*
iot:EnableTopicRule	EnableTopicRule	arn:aws:iot:*region*:*account-id*:rule/*rule-name*
iot:GetEffectivePolicies	GetEffectivePolicies	arn:aws:iot:*region*:*account-id*:cert/*cert-id*
iot:GetIndexingConfiguration	GetIndexingConfiguration	Nenhum
iot:GetJobDocument	GetJobDocument	arn:aws:iot:*region*:*account-id*:job/*job-id*
iot:GetLoggingOptions	GetLoggingOptions	*
iot:GetPolicy	GetPolicy	arn:aws:iot:*region*:*account-id*:policy/*policy-name*
iot:GetPolicyVersion	GetPolicyVersion	arn:aws:iot:*region*:*account-id*:policy/*policy-name*
iot:GetRegistrationCode	GetRegistrationCode	*
iot:GetTopicRule	GetTopicRule	arn:aws:iot:*region*:*account-id*:rule/*rule-name*

Ações de políticas	AWS IoT API	Recursos
iot:ListAttachedPolicies	ListAttachedPolicies	`arn:aws:iot:`*`region`*`:`*`account-id`*`:thinggroup/`*`thing-group-name`* ou `arn:aws:iot:`*`region`*`:`*`account-id`*`:cert/`*`cert-id`*
iot:ListAuthorizers	ListAuthorizers	Nenhum
iot:ListCACertificates	ListCACertificates	*
iot:ListCertificates	ListCertificates	*
iot:ListCertificatesByCA	ListCertificatesByCA	*
iot:ListIndices	ListIndices	Nenhum
iot:ListJobExecutionsForJob	ListJobExecutionsForJob	Nenhum
iot:ListJobExecutionsForThing	ListJobExecutionsForThing	Nenhum
iot:ListJobs	ListJobs	`arn:aws:iot:`*`region`*`:`*`account-id`*`:thinggroup/`*`thing-group-name`* E se thingGroupName parâmetro usado
IOT:ListJobTemplates	ListJobs	Nenhum
iot:ListOutgoingCertificates	ListOutgoingCertificates	*
iot:ListPolicies	ListPolicies	*
iot:ListPolicyPrincipals	ListPolicyPrincipals	`arn:aws:iot:`*`region`*`:`*`account-id`*`:policy/`*`policy-name`*
iot:ListPolicyVersions	ListPolicyVersions	`arn:aws:iot:`*`region`*`:`*`account-id`*`:policy/`*`policy-name`*
iot:ListPrincipalPolicies	ListPrincipalPolicies	`arn:aws:iot:`*`region`*`:`*`account-id`*`:cert/`*`cert-id`*
iot:ListPrincipalThings	ListPrincipalThings	`arn:aws:iot:`*`region`*`:`*`account-id`*`:cert/`*`cert-id`*
iot:ListRoleAliases	ListRoleAliases	Nenhum
iot:ListTargetsForPolicy	ListTargetsForPolicy	`arn:aws:iot:`*`region`*`:`*`account-id`*`:policy/`*`policy-name`*
iot:ListThingGroups	ListThingGroups	Nenhum
iot:ListThingGroupsForThing	ListThingGroupsForThing	`arn:aws:iot:`*`region`*`:`*`account-id`*`:thing/`*`thing-name`*
iot:ListThingPrincipals	ListThingPrincipals	`arn:aws:iot:`*`region`*`:`*`account-id`*`:thing/`*`thing-name`*
iot:ListThingRegistrationTaskReports	ListThingRegistrationTaskReports	Nenhum
iot:ListThingRegistrationTasks	ListThingRegistrationTasks	Nenhum
iot:ListThingTypes	ListThingTypes	*
iot:ListThings	ListThings	*

Ações de políticas	AWS IoT API	Recursos
iot:ListThingsInThingGroup	ListThingsInThingGroup	arn:aws:iot:*region*:*account-id*:thinggroup/*thing-group-name*
iot:ListTopicRules	ListTopicRules	*
iot:ListV2LoggingLevels	ListV2LoggingLevels	Nenhum
iot:RegisterCACertificate	RegisterCACertificate	*
iot:RegisterCertificate	RegisterCertificate	*
iot:RegisterThing	RegisterThing	Nenhum
iot:RejectCertificateTransfer	RejectCertificateTransfer	arn:aws:iot:*region*:*account-id*:cert/*cert-id*
iot:RemoveThingFromThingGroup	RemoveThingFromThingGroup	arn:aws:iot:*region*:*account-id*:thinggroup/*thing-group-name* arn:aws:iot:*region*:*account-id*:thing/*thing-name*
iot:ReplaceTopicRule	ReplaceTopicRule	arn:aws:iot:*region*:*account-id*:rule/*rule-name*
iot:SearchIndex	SearchIndex	arn:aws:iot:*region*:*account-id*:index/*index-id*
iot:SetDefaultAuthorizer	SetDefaultAuthorizer	arn:aws:iot:*region*:*account-id*:authorizer/*authorizer-function-name*
iot:SetDefaultPolicyVersion	SetDefaultPolicyVersion	arn:aws:iot:*region*:*account-id*:policy/*policy-name*
iot:SetLoggingOptions	SetLoggingOptions	arn:aws:iot:*region*:*account-id*:role/*role-name*
iot:SetV2LoggingLevel	SetV2LoggingLevel	arn:aws:iot:*region*:*account-id*:thinggroup/*thing-group-name*
iot:SetV2LoggingOptions	SetV2LoggingOptions	arn:aws:iot:*region*:*account-id*:role/*role-name*
iot:StartThingRegistrationTask	StartThingRegistrationTask	Nenhum
iot:StopThingRegistrationTask	StopThingRegistrationTask	Nenhum
iot:TestAuthorization	TestAuthorization	arn:aws:iot:*region*:*account-id*:cert/*cert-id*
iot:TestInvokeAuthorizer	TestInvokeAuthorizer	Nenhum
iot:TransferCertificate	TransferCertificate	arn:aws:iot:*region*:*account-id*:cert/*cert-id*
iot:UpdateAuthorizer	UpdateAuthorizer	arn:aws:iot:*region*:*account-id*:authorizerfunction/*authorizer-function-name*
iot:UpdateCACertificate	UpdateCACertificate	arn:aws:iot:*region*:*account-id*:cacert/*cert-id*
iot:UpdateCertificate	UpdateCertificate	arn:aws:iot:*region*:*account-id*:cert/*cert-id*
iot:UpdateDimension	UpdateDimension	arn:aws:iot:*region*:*account-id*:dimension/*dimension-name*
iot:UpdateEventConfigurations	UpdateEventConfigurations	Nenhum
iot:UpdateIndexingConfiguration	UpdateIndexingConfiguration	Nenhum

Ações de políticas	AWS IoT API	Recursos
iot:UpdateRoleAlias	UpdateRoleAlias	`arn:aws:iot:region:account-id:rolealias/role-alias-name`

iot:UpdateSecurityProfile	UpdateSecurityProfile	`arn:aws:iot:region:account-id:securityprofile/security-profile-name`
		
		`arn:aws:iot:region:account-id:dimension/dimension-name`

iot:UpdateThing	UpdateThing	`arn:aws:iot:region:account-id:thing/thing-name`
iot:UpdateThingGroup	UpdateThingGroup	`arn:aws:iot:region:account-id:thinggroup/thing-group-name`
iot:UpdateThingGroupsForThing	UpdateThingGroupsForThing	`arn:aws:iot:region:account-id:thing/thing-name`

Ações de políticas noAWS IoTUse o seguinte prefixo antes da ação: `iot:`. Por exemplo, para conceder permissão a alguém para listar todas as coisas do IoT registradas em seusConta da AWScom o`ListThings`API, você inclui `oiot:ListThings`ação em sua política. As declarações de política devem incluir um elemento `Action` ou `NotAction`. O AWS IoT define seu próprio conjunto de ações que descrevem as tarefas que podem ser executadas com esse serviço.

Para especificar várias ações em uma única declaração, separe-as com vírgulas, conforme o seguinte:

```
"Action": [
      "ec2:action1",
      "ec2:action2"
```

Você também pode especificar várias ações usando caracteres curinga (*). Por exemplo, para especificar todas as ações que começam com a palavra `Describe`, inclua a seguinte ação:

```
"Action": "iot:Describe*"
```

Para ver uma lista de ações do AWS IoT, consulte Ações definidas pelo AWS IoT no Manual do usuário do IAM.

Ações do Device Advisor

A tabela a seguir lista as ações do IAM IoT Device Advisor, as associadasAWS IoTA API do Device Advisor e o recurso que a ação manipula.

Ações de políticas	AWS IoT API	Recursos
IOTDeviceAdvisor:CreateSuiteDefinition	CreateSuiteDefinition	Nenhum
IOTDeviceAdvisor:DeleteSuiteDefinition	DeleteSuiteDefinition	`arn:aws:iotdeviceadvisor:region:account-id:suitedefinition/suite-definition-id`
IOTDeviceAdvisor:GetSuiteDefinition	GetSuiteDefinition	`arn:aws:iotdeviceadvisor:region:account-id:suitedefinition/suite-definition-id`
IOTDeviceAdvisor:GetSuiteRun	GetSuiteRun	`arn:aws:iotdeviceadvisor:region:account-id:suitedefinition/suite-run-id`

Ações de políticas	AWS IoT API	Recursos
IOTDeviceAdvisor:GetSuiteRunReport	GetSuiteRunReport	`arn:aws:iotdeviceadvisor:`*`region`*`:`*`account-id`*`:suiterun/`*`suite-definition-id`*`/`*`suite-run-id`
IOTDeviceAdvisor:ListSuiteDefinitions	ListSuiteDefinitions	Nenhum
IOTDeviceAdvisor:ListSuiteRuns	ListSuiteRuns	`arn:aws:iotdeviceadvisor:`*`region`*`:`*`account-id`*`:suitedefinition/`*`suite-definition-id`

IOTDeviceAdvisor:ListTagsForResource	ListTagsForResource	`arn:aws:iotdeviceadvisor:`*`region`*`:`*`account-id`*`:suitedefinition/`*`suite-definition-id`
		
		`arn:aws:iotdeviceadvisor:`*`region`*`:`*`account-id`*`:suiterun/suite-definition-id/`*`suite-run-id`

IOTDeviceAdvisor:StartSuiteRun	StartSuiteRun	`arn:aws:iotdeviceadvisor:`*`region`*`:`*`account-id`*`:suitedefinition/`*`suite-definition-id`

IOTDeviceAdvisor:TagResource	TagResource	`arn:aws:iotdeviceadvisor:`*`region`*`:`*`account-id`*`:suitedefinition/`*`suite-definition-id`
		
		`arn:aws:iotdeviceadvisor:`*`region`*`:`*`account-id`*`:suiterun/suite-definition-id/`*`suite-run-id`

IOTDeviceAdvisor:UntagResource	UntagResource	`arn:aws:iotdeviceadvisor:`*`region`*`:`*`account-id`*`:suitedefinition/`*`suite-definition-id`
		
		`arn:aws:iotdeviceadvisor:`*`region`*`:`*`account-id`*`:suiterun/suite-definition-id/`*`suite-run-id`

IOTDeviceAdvisor:UpdateSuiteDefinition	UpdateSuiteDefinition	`arn:aws:iotdeviceadvisor:`*`region`*`:`*`account-id`*`:suitedefinition/`*`suite-definition-id`
IOTDeviceAdvisor:StopSuiteRun	StopSuiteRun	`arn:aws:iotdeviceadvisor:`*`region`*`:`*`account-id`*`:suiterun/suite-definition-id/`*`suite-run-id`

Ações de políticas noAWS IoTO Supervisor de dispositivos usa o seguinte prefixo antes da ação: `iotdeviceadvisor:`. Por exemplo, para conceder permissão a alguém para listar todas as definições de suíte registradas em seusConta da AWScom o ListSuiteDefinitions API, você inclui `oiotdeviceadvisor:ListSuiteDefinitions`ação em sua política.

Recursos

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

O elemento `Resource` de política JSON especifica o objeto ou os objetos aos quais a ação se aplica. As instruções devem incluir um elemento `Resource` ou um elemento `NotResource`. Como prática recomendada, especifique um recurso usando seu Nome de recurso da Amazon (ARN). Isso pode ser feito para ações que oferecem suporte a um tipo de recurso específico, conhecido como permissões em nível de recurso.

Para ações que não oferecem suporte a permissões em nível de recurso, como operações de listagem, use um curinga (*) para indicar que a instrução se aplica a todos os recursos.

```
"Resource": "*"
```

Recursos do AWS IoT

Ações de políticas	AWS IoT API	Recursos
iot:AcceptCertificateTransfer	AcceptCertificateTransfer	`arn:aws:iot:region:account-id:cert/cert-id` **Note** OConta da AWSEspecificada no ARN deve ser a conta para a qual o certificado está sendo transferido.
iot:AddThingToThingGroup	AddThingToThingGroup	`arn:aws:iot:region:account-id:thinggroup/thing-group-name` `arn:aws:iot:region:account-id:thing/thing-name`
iot:AssociateTargetsWithJob	AssociateTargetsWithJob	Nenhum

iot:AttachPolicy	AttachPolicy	`arn:aws:iot:region:account-id:thinggroup/thing-group-name`
		ou
		`arn:aws:iot:region:account-id:cert/cert-id`

iot:AttachPrincipalPolicy	AttachPrincipalPolicy	`arn:aws:iot:region:account-id:cert/cert-id`
iot:AttachThingPrincipal	AttachThingPrincipal	`arn:aws:iot:region:account-id:cert/cert-id`
iot:CancelCertificateTransfer	CancelCertificateTransfer	`arn:aws:iot:region:account-id:cert/cert-id` **Note** OConta da AWSEspecificada no ARN deve ser a conta para a qual o certificado está sendo transferido.
iot:CancelJob	CancelJob	`arn:aws:iot:region:account-id:job/job-id`
iot:CancelJobExecution	CancelJobExecution	`arn:aws:iot:region:account-id:job/job-id` `arn:aws:iot:region:account-id:thing/thing-name`
iot:ClearDefaultAuthorizer	ClearDefaultAuthorizer	Nenhum
iot:CreateAuthorizer	CreateAuthorizer	`arn:aws:iot:region:account-id:authorizer/authorizer-function-name`
iot:CreateCertificateFromCsr	CreateCertificateFromCsr	*

iot:CreateJob	CreateJob	`arn:aws:iot:region:account-id:job/job-id`
		`arn:aws:iot:region:account-id:thinggroup/thing-group-name`
		`arn:aws:iot:region:account-id:thing/thing-name`
		`arn:aws:iot:region:account-id:jobtemplate/job-template-id`

IOT: CreateJOBTemplate	CreateJobTemplate	`arn:aws:iot:region:account-id:job/job-id`

Ações de políticas	AWS IoT API	Recursos
		`arn:aws:iot:`*region*`:`*account-id*`:jobtemplate/`*job-template-id*
iot:CreateKeysAndCertificate	CreateKeysAndCertificate	*
iot:CreatePolicy	CreatePolicy	`arn:aws:iot:`*region*`:`*account-id*`:policy/`*policy-name*
CreatePolicyVersion	iot:CreatePolicyVersion	`arn:aws:iot:`*region*`:`*account-id*`:policy/`*policy-name* Note Isso deve ser umAWS IoTpolítica, não uma política do IAM.
iot:CreateRoleAlias	CreateRoleAlias	(parâmetro: roleAlias) `arn:aws:iot:`*region*`:`*account-id*`:rolealias/`*role-alias-name*
iot:CreateThing	CreateThing	`arn:aws:iot:`*region*`:`*account-id*`:thing/`*thing-name*
iot:CreateThingGroup	CreateThingGroup	`arn:aws:iot:`*region*`:`*account-id*`:thinggroup/`*thing-group-name* para o grupo que está sendo criado e para o grupo pai, se usado
iot:CreateThingType	CreateThingType	`arn:aws:iot:`*region*`:`*account-id*`:thingtype/`*thing-type-name*
iot:CreateTopicRule	CreateTopicRule	`arn:aws:iot:`*region*`:`*account-id*`:rule/`*rule-name*
iot:DeleteAuthorizer	DeleteAuthorizer	`arn:aws:iot:`*region*`:`*account-id*`:authorizer/`*authorizer-name*
iot:DeleteCACertificate	DeleteCACertificate	`arn:aws:iot:`*region*`:`*account-id*`:cacert/`*cert-id*
iot:DeleteCertificate	DeleteCertificate	`arn:aws:iot:`*region*`:`*account-id*`:cert/`*cert-id*
iot:DeleteJob	DeleteJob	`arn:aws:iot:`*region*`:`*account-id*`:job/`*job-id*
iot:DeleteJobExecution	DeleteJobExecution	`arn:aws:iot:`*region*`:`*account-id*`:job/`*job-id* `arn:aws:iot:`*region*`:`*account-id*`:thing/`*thing-name*
IOT: DeleteJobTemplate	DeleteJobTemplate	`arn:aws:iot:`*region*`:`*account-id*`:jobtemplate/`*job-template-id*
iot:DeletePolicy	DeletePolicy	`arn:aws:iot:`*region*`:`*account-id*`:policy/`*policy-name*
iot:DeletePolicyVersion	DeletePolicyVersion	`arn:aws:iot:`*region*`:`*account-id*`:policy/`*policy-name*
iot:DeleteRegistrationCode	DeleteRegistrationCode	*
iot:DeleteRoleAlias	DeleteRoleAlias	`arn:aws:iot:`*region*`:`*account-id*`:rolealias/`*role-alias-name*

Ações de políticas	AWS IoT API	Recursos
iot:DeleteThing	DeleteThing	arn:aws:iot:*region*:*account-id*:thing/*thing-name*
iot:DeleteThingGroup	DeleteThingGroup	arn:aws:iot:*region*:*account-id*:thinggroup/*thing-group-name*
iot:DeleteThingType	DeleteThingType	arn:aws:iot:*region*:*account-id*:thingtype/*thing-type-name*
iot:DeleteTopicRule	DeleteTopicRule	arn:aws:iot:*region*:*account-id*:rule/*rule-name*
iot:DeleteV2LoggingLevel	DeleteV2LoggingLevel	arn:aws:iot:*region*:*account-id*:thinggroup/*thing-group-name*
iot:DeprecateThingType	DeprecateThingType	arn:aws:iot:*region*:*account-id*:thingtype/*thing-type-name*
iot:DescribeAuthorizer	DescribeAuthorizer	arn:aws:iot:*region*:*account-id*:authorizer/*authorizer-function-name* (parâmetro: authorizerName) none
iot:DescribeCACertificate	DescribeCACertificate	arn:aws:iot:*region*:*account-id*:cacert/*cert-id*
iot:DescribeCertificate	DescribeCertificate	arn:aws:iot:*region*:*account-id*:cert/*cert-id*
iot:DescribeDefaultAuthorizer	DescribeDefaultAuthorizer	Nenhum
iot:DescribeEndpoint	DescribeEndpoint	*
iot:DescribeEventConfigurations	DescribeEventConfigurations	*
iot:DescribeIndex	DescribeIndex	arn:aws:iot:*region*:*account-id*:index/*index-name*
iot:DescribeJob	DescribeJob	arn:aws:iot:*region*:*account-id*:job/*job-id*
iot:DescribeJobExecution	DescribeJobExecution	Nenhum
IOT:DescribeJobTemplate	DescribeJobTemplate	arn:aws:iot:*region*:*account-id*:jobtemplate/*job-template-id*
iot:DescribeRoleAlias	DescribeRoleAlias	arn:aws:iot:*region*:*account-id*:rolealias/*role-alias-name*
iot:DescribeThing	DescribeThing	arn:aws:iot:*region*:*account-id*:thing/*thing-name*
iot:DescribeThingGroup	DescribeThingGroup	arn:aws:iot:*region*:*account-id*:thinggroup/*thing-group-name*
iot:DescribeThingRegistrationTask	DescribeThingRegistrationTask	Nenhum
iot:DescribeThingType	DescribeThingType	arn:aws:iot:*region*:*account-id*:thingtype/*thing-type-name*
iot:DetachPolicy	DetachPolicy	arn:aws:iot:*region*:*account-id*:cert/*cert-id* ou arn:aws:iot:*region*:*account-id*:thinggroup/*thing-group-name*

Ações de políticas	AWS IoT API	Recursos
iot:DetachPrincipalPolicy	DetachPrincipalPolicy	`arn:aws:iot:region:account-id:cert/cert-id`
iot:DetachThingPrincipal	DetachThingPrincipal	`arn:aws:iot:region:account-id:cert/cert-id`
iot:DisableTopicRule	DisableTopicRule	`arn:aws:iot:region:account-id:rule/rule-name`
iot:EnableTopicRule	EnableTopicRule	`arn:aws:iot:region:account-id:rule/rule-name`
iot:GetEffectivePolicies	GetEffectivePolicies	`arn:aws:iot:region:account-id:cert/cert-id`
iot:GetIndexingConfiguration	GetIndexingConfiguration	Nenhum
iot:GetJobDocument	GetJobDocument	`arn:aws:iot:region:account-id:job/job-id`
iot:GetLoggingOptions	GetLoggingOptions	*
iot:GetPolicy	GetPolicy	`arn:aws:iot:region:account-id:policy/policy-name`
iot:GetPolicyVersion	GetPolicyVersion	`arn:aws:iot:region:account-id:policy/policy-name`
iot:GetRegistrationCode	GetRegistrationCode	*
iot:GetTopicRule	GetTopicRule	`arn:aws:iot:region:account-id:rule/rule-name`

iot:ListAttachedPolicies	ListAttachedPolicies	`arn:aws:iot:region:account-id:thinggroup/thing-group-name`
		ou
		`arn:aws:iot:region:account-id:cert/cert-id`

iot:ListAuthorizers	ListAuthorizers	Nenhum
iot:ListCACertificates	ListCACertificates	*
iot:ListCertificates	ListCertificates	*
iot:ListCertificatesByCA	ListCertificatesByCA	*
iot:ListIndices	ListIndices	Nenhum
iot:ListJobExecutionsForJob	ListJobExecutionsForJob	Nenhum
iot:ListJobExecutionsForThing	ListJobExecutionsForThing	Nenhum
iot:ListJobs	ListJobs	`arn:aws:iot:region:account-id:thinggroup/thing-group-name` E se thingGroupName parâmetro usado
IOT:ListJobTemplates	ListJobTemplates	Nenhum
iot:ListOutgoingCertificates	ListOutgoingCertificates	*
iot:ListPolicies	ListPolicies	*
iot:ListPolicyPrincipals	ListPolicyPrincipals	`arn:aws:iot:region:account-id:policy/policy-name`

Ações de políticas	AWS IoT API	Recursos
iot:ListPolicyVersions	ListPolicyVersions	arn:aws:iot:*region*:*account-id*:policy/*policy-name*
iot:ListPrincipalPolicies	ListPrincipalPolicies	arn:aws:iot:*region*:*account-id*:cert/*cert-id*
iot:ListPrincipalThings	ListPrincipalThings	arn:aws:iot:*region*:*account-id*:cert/*cert-id*
iot:ListRoleAliases	ListRoleAliases	Nenhum
iot:ListTargetsForPolicy	ListTargetsForPolicy	arn:aws:iot:*region*:*account-id*:policy/*policy-name*
iot:ListThingGroups	ListThingGroups	Nenhum
iot:ListThingGroupsForThing	ListThingGroupsForThing	arn:aws:iot:*region*:*account-id*:thing/*thing-name*
iot:ListThingPrincipals	ListThingPrincipals	arn:aws:iot:*region*:*account-id*:thing/*thing-name*
iot:ListThingRegistrationTaskReports	ListThingRegistrationTaskReports	Nenhum
iot:ListThingRegistrationTasks	ListThingRegistrationTasks	Nenhum
iot:ListThingTypes	ListThingTypes	*
iot:ListThings	ListThings	*
iot:ListThingsInThingGroup	ListThingsInThingGroup	arn:aws:iot:*region*:*account-id*:thinggroup/*thing-group-name*
iot:ListTopicRules	ListTopicRules	*
iot:ListV2LoggingLevels	ListV2LoggingLevels	Nenhum
iot:RegisterCACertificate	RegisterCACertificate	*
iot:RegisterCertificate	RegisterCertificate	*
iot:RegisterThing	RegisterThing	Nenhum
iot:RejectCertificateTransfer	RejectCertificateTransfer	arn:aws:iot:*region*:*account-id*:cert/*cert-id*
iot:RemoveThingFromThingGroup	RemoveThingFromThingGroup	arn:aws:iot:*region*:*account-id*:thinggroup/*thing-group-name* arn:aws:iot:*region*:*account-id*:thing/*thing-name*
iot:ReplaceTopicRule	ReplaceTopicRule	arn:aws:iot:*region*:*account-id*:rule/*rule-name*
iot:SearchIndex	SearchIndex	arn:aws:iot:*region*:*account-id*:index/*index-id*
iot:SetDefaultAuthorizer	SetDefaultAuthorizer	arn:aws:iot:*region*:*account-id*:authorizer/*authorizer-function-name*
iot:SetDefaultPolicyVersion	SetDefaultPolicyVersion	arn:aws:iot:*region*:*account-id*:policy/*policy-name*
iot:SetLoggingOptions	SetLoggingOptions	arn:aws:iot:*region*:*account-id*:role/*role-name*
iot:SetV2LoggingLevel	SetV2LoggingLevel	arn:aws:iot:*region*:*account-id*:thinggroup/*thing-group-name*

Ações de políticas	AWS IoT API	Recursos
iot:SetV2LoggingOptions	SetV2LoggingOptions	`arn:aws:iot:region:account-id:role/role-name`
iot:StartThingRegistrationTask	StartThingRegistrationTask	Nenhum
iot:StopThingRegistrationTask	StopThingRegistrationTask	Nenhum
iot:TestAuthorization	TestAuthorization	`arn:aws:iot:region:account-id:cert/cert-id`
iot:TestInvokeAuthorizer	TestInvokeAuthorizer	Nenhum
iot:TransferCertificate	TransferCertificate	`arn:aws:iot:region:account-id:cert/cert-id`
iot:UpdateAuthorizer	UpdateAuthorizer	`arn:aws:iot:region:account-id:authorizerfunction/authorizer-function-name`
iot:UpdateCACertificate	UpdateCACertificate	`arn:aws:iot:region:account-id:cacert/cert-id`
iot:UpdateCertificate	UpdateCertificate	`arn:aws:iot:region:account-id:cert/cert-id`
iot:UpdateEventConfigurations	UpdateEventConfigurations	Nenhum
iot:UpdateIndexingConfiguration	UpdateIndexingConfiguration	Nenhum
iot:UpdateRoleAlias	UpdateRoleAlias	`arn:aws:iot:region:account-id:rolealias/role-alias-name`
iot:UpdateThing	UpdateThing	`arn:aws:iot:region:account-id:thing/thing-name`
iot:UpdateThingGroup	UpdateThingGroup	`arn:aws:iot:region:account-id:thinggroup/thing-group-name`
iot:UpdateThingGroupsForThing	UpdateThingGroupsForThing	`arn:aws:iot:region:account-id:thing/thing-name`

Para obter mais informações sobre o formato de ARNs, consulte Nomes de recursos da Amazon (ARNs) e namespaces de serviços da AWS.

Algumas ações do AWS IoT, como as ações para a criação de recursos, não podem ser executadas em um recurso específico. Nesses casos, você deve usar o caractere curinga (*).

```
"Resource": "*"
```

Para ver uma lista deAWS IoTOs tipos de recurso e seus ARNs, consulteRecursos definidos peloAWS IoTnoManual do usuário do IAM. Para saber com quais ações você pode especificar o ARN de cada recurso, consulte Ações definidas pelo AWS IoT.

Recursos do Device Advisor

Para definir restrições no nível do recurso paraAWS IoTPolíticas do IAM do Device Advisor, use os seguintes formatos ARN de recursos para definições de suíte e execuções de suítes.

Formato do ARN de recurso de definição de conjunto

```
arn:aws:iotdeviceadvisor:region:account-id:suitedefinition/suite-definition-id
```

Formato do ARN de recurso de execução do pacote

```
arn:aws:iotdeviceadvisor:region:account-id:suiterun/suite-definition-id/suite-run-id
```

Chaves de condição

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

O elemento `Condition` (ou bloco de `Condition`) permite que você especifique condições nas quais uma instrução está em vigor. O elemento `Condition` é opcional. É possível criar expressões condicionais que usam agentes 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 elemento `Condition`, 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 para que as permissões da instrução sejam concedidas.

Você também pode usar variáveis de espaço reservado ao especificar as condições. Por exemplo, é possível conceder a um usuário do IAM permissão para acessar um recurso somente se ele estiver marcado com seu nome de usuário do IAM. Para obter mais informações, consulte Elementos de política do IAM: variáveis e etiquetas no Guia do usuário do IAM.

A AWS oferece suporte a chaves de condição globais e chaves de condição específicas do serviço. Para ver todas as chaves de condição globais da AWS, consulte Chaves de contexto de condição globais da AWS no Guia do usuário do IAM.

O AWS IoT define seu próprio conjunto de chaves de condição e também oferece suporte ao uso de algumas chaves de condição globais. Para ver todas as chaves de condição globais da AWS, consulte Chaves de contexto de condição globais da AWS no Manual do usuário do IAM.

Chaves de condição do AWS IoT

Chaves de condição do AWS IoT	Descrição	Type
`aws:RequestTag/${tag-key}`	Uma chave de tag que está presente na solicitação que o usuário faz para o AWS IoT.	String
`aws:ResourceTag/${tag-key}`	O componente da chave de uma tag anexada a um recurso do AWS IoT.	String
`aws:TagKeys`	A lista de todos os nomes de chaves de etiquetas associadas ao recurso na solicitação.	String

Para ver uma lista deAWS IoTChaves de condição, consulteChaves de condição doAWS IoTnoManual do usuário do IAM. Para saber com quais ações e recursos é possível usar uma chave de condição, consulte Ações definidas pelo AWS IoT.

Exemplos

Para visualizar exemplos de políticas baseadas em identidade do AWS IoT, consulte Exemplos de políticas baseadas em identidade do AWS IoT (p. 404).

Políticas baseadas em recursos do AWS IoT

As políticas baseadas em recursos são documentos de políticas JSON que especificam quais ações uma entidade principal pode executar no recurso do AWS IoT e sob quais condições.

AWS IoTO não oferece suporte a políticas baseadas em recurso do IAM. No entanto, ele suportaAWS IoTPolíticas com base em recurso do . Para obter mais informações, consultePolíticas do AWS IoT Core (p. 332)

Autorização baseada em tags do AWS IoT

Você pode anexar tags a recursos do AWS IoT ou passar tags em uma solicitação ao AWS IoT. Para controlar o acesso baseado em tags, forneça informações sobre as tags no elemento de condição de uma política usando as `iot:ResourceTag/key-name`, `aws:RequestTag/key-name` ou `aws:TagKeys` chaves de condição. Para obter mais informações, consulteUtilização de tags com políticas do IAM (p. 289) Para obter mais informações sobre recursos de marcação do AWS IoT, consulte Marcar recursos do AWS IoT (p. 288).

Para visualizar um exemplo de política baseada em identidade para limitar o acesso a um recurso baseado em tags desse recurso, consulte Visualização de recursos do AWS IoT com base em tags (p. 406).

Funções do IAM no AWS IoT

Uma função do IAM é uma entidade dentro da sua Conta da AWS que tem permissões específicas.

Usar credenciais temporárias com o AWS IoT

É possível usar credenciais temporárias para fazer login com federação, assumir um perfil do IAM ou assumir um perfil entre contas. As credenciais de segurança temporárias são obtidas chamando operações da API do AWS STS, como AssumeRole ou GetFederationToken.

O AWS IoT oferece suporte ao uso de credenciais temporárias.

Funções vinculadas ao serviço

Funções vinculadas ao serviço permitem que os serviços da AWS acessem recursos em outros serviços para concluir uma ação em seu nome. As funções vinculadas ao serviço aparecem em sua conta do IAM e são de propriedade do serviço. Um administrador do IAM pode visualizar, mas não pode editar as permissões para funções vinculadas ao serviço.

O AWS IoT não oferece suporte às funções vinculadas ao serviço.

Perfis de serviço

Esse recurso permite que um serviço assuma um perfil de serviço em seu nome. A função permite que o serviço acesse recursos em outros serviços para concluir uma ação em seu nome. As funções de serviço aparecem em sua conta do IAM e são de propriedade da conta. Isso significa que um administrador do IAM pode alterar as permissões para essa função. Porém, fazer isso pode alterar a funcionalidade do serviço.

Exemplos de políticas baseadas em identidade do AWS IoT

Por padrão, os usuários e as funções do IAM não têm permissão para criar ou modificar recursos do AWS IoT. Eles também não podem executar tarefas usando o AWS Management Console, a AWS CLI ou uma

API da AWS. Um administrador do IAM deve criar políticas do IAM que concedam aos usuários e perfis permissão para executarem operações de API específicas nos recursos especificados de que precisam. O administrador deve anexar essas políticas aos usuários ou grupos do IAM que exigem essas permissões.

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

Tópicos

- Práticas recomendadas de políticas (p. 405)
- Usar o console do AWS IoT (p. 405)
- Permitir que os usuários visualizem suas próprias permissões (p. 406)
- Visualização de recursos do AWS IoT com base em tags (p. 406)
- Visualizar oAWS IoTRecursos do Device Advisor com base em tags (p. 407)

Práticas recomendadas de políticas

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

- Primeiro, use políticas gerenciadas pela AWS: para começar a usar o AWS IoT rapidamente, use as políticas gerenciadas pela AWS para conceder a 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 Começar a usar permissões com políticas gerenciadas da AWS 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 MFA para operações confidenciais: para aumentar a segurança, exija que os usuários do IAM usem Multi-Factor Authentication (MFA) para acessar recursos ou operações de API confidenciais. Para obter mais informações, consulte Usar autenticação multifator (MFA) 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, consulteElementos de política JSON do IAM: CondiçãonoManual do usuário do IAM.

Usar o console do AWS IoT

Para acessar o console do AWS IoT, você deve ter um conjunto mínimo de permissões. Essas permissões devem permitir que você liste e visualize detalhes doAWS IoTRecursos noConta 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 funções do IAM) com essa política.

Para garantir que essas entidades ainda possam usar o console do AWS IoT, anexe também a seguinte política gerenciada pela AWS às entidades: `AWSIoTFullAccess`. Para obter mais informações, consulteComo adicionar permissões a um usuárionoManual do usuário do IAM.

Não é necessário conceder permissões mínimas do console para usuários que fazem chamadas somente à AWS CLI ou à API do AWS. 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 do IAM visualizem as políticas gerenciadas e em linha anexadas a sua identidade de usuário. Essa política inclui permissões para concluir essa ação no console ou de forma programática usando a AWS CLI ou a API da 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": "*"
        }
    ]
}
```

Visualização de recursos do AWS IoT com base em tags

Você pode usar condições em sua política baseada em identidade para controlar o acesso aos recursos do AWS IoT com base em tags. Este exemplo mostra como é possível criar uma política que permite visualizar uma coisa. No entanto, a permissão é concedida somente se a tag da coisa `Owner` tiver o valor do nome de usuário desse usuário. Essa política também concede as permissões necessárias concluir essa ação no console.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ListBillingGroupsInConsole",
            "Effect": "Allow",
            "Action": "iot:ListBillingGroups",
            "Resource": "*"
        },
        {
            "Sid": "ViewBillingGroupsIfOwner",
            "Effect": "Allow",
           "Action": "iot:DescribeBillingGroup",
            "Resource": "arn:aws:iot:*:*:billinggroup/*",
```

```
            "Condition": {
                "StringEquals": {"aws:ResourceTag/Owner": "${aws:username}"}
            }
        }
    ]
}
```

É possível anexar essa política aos usuários do IAM na sua conta. Se um usuário nomeado `richard-roe` tentar exibir um grupo de faturamento AWS IoT, o grupo de faturamento deve ser marcado com tags `Owner=richard-roe` ou `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 fazem distinção entre maiúsculas e minúsculas. Para obter mais informações, consulteElementos de política JSON do IAM: CondiçãonoManual do usuário do IAM.

Visualizar oAWS IoTRecursos do Device Advisor com base em tags

Você pode usar condições em sua política baseada em identidade para controlar o acesso aoAWS IoTRecursos do Device Advisor com base em tags. O exemplo a seguir mostra como você pode criar uma política que permite visualizar uma definição de suíte específica. No entanto, a permissão será concedida somente se a tag de definição de suíte tiver`SuiteType`Defina o valor de`MQTT`. Essa política também concede as permissões necessárias concluir essa ação no console.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewSuiteDefinition",
            "Effect": "Allow",
            "Action": "iotdeviceadvisor:GetSuiteDefinition",
            "Resource": "arn:aws:iotdeviceadvisor:*:*:suitedefinition/*",
            "Condition": {
                "StringEquals": {"aws:ResourceTag/SuiteType": "MQTT"}
            }
        }
    ]
}
```

Políticas gerenciadas pela AWS para o AWS IoT

Para adicionar permissões a usuários, grupos e funções, é mais fácil usar políticas gerenciadas pela AWS do que gravar políticas por conta própria. É necessário tempo e experiência para criar políticas gerenciadas pelo cliente do IAM que fornecem à sua equipe apenas as permissões de que precisam. Para começar rapidamente, você pode usar nossas políticas gerenciadas pela AWS. Essas políticas abrangem casos de uso comuns e estão disponíveis na sua Conta da AWS. Para obter mais informações sobre as políticas gerenciadas da AWS, consulte Políticas gerenciadas da AWS no Guia do usuário do IAM.

Os serviços da AWS mantêm e atualizam políticas gerenciadas pela AWS. Não é possível alterar as permissões em políticas gerenciadas pela AWS. Os serviços ocasionalmente acrescentam permissões adicionais a uma política gerenciada pela AWS para oferecer suporte a novos recursos. Esse tipo de atualização afeta todas as identidades (usuários, grupos e funções) em que a política está anexada. É mais provável que os serviços atualizem uma política gerenciada pela AWS quando um novo recurso for iniciado ou novas operações se tornarem disponíveis. Os serviços não removem permissões de uma política gerenciada pela AWS, portanto, as atualizações de políticas não suspendem suas permissões existentes.

Além disso, a AWS oferece suporte a políticas gerenciadas para funções de trabalho que abrangem vários serviços. Por exemplo, a política gerenciada pela AWS denominada ReadOnlyAccess fornece acesso somente leitura a todos os serviços e recursos da AWS. Quando um serviço executa um novo recurso, a AWS adiciona permissões somente leitura para novas operações e recursos. Para obter uma lista e descrições das políticas de funções de trabalho, consulte Políticas gerenciadas pela AWS para funções de trabalho no Guia do usuário do IAM.

> **Note**
>
> AWS IoTfunciona com ambosAWS IoTe políticas do IAM. Este tópico discute somente políticas do IAM, o que define uma ação de política para operações de API de plano de controle e plano de dados. Consulte também Políticas do AWS IoT Core (p. 332).

AWSpolítica gerenciada: AWSIoTConfigAccess

É possível anexar a política `AWSIoTConfigAccess` a suas identidades do IAM.

Essa política concede permissões de identidade associadas que permitem acesso a todosAWS IoToperações de configuração. Essa política pode afetar o processamento de dados e armazenamento. Para exibir esta política noAWS Management ConsoleConsulte,AWSIoTConfigAccess.

Detalhes da permissão

Esta política inclui as seguintes permissões.

- `iot`— RecuperarAWS IoTdados e execute ações de configuração de IoT.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:AcceptCertificateTransfer",
                "iot:AddThingToThingGroup",
                "iot:AssociateTargetsWithJob",
                "iot:AttachPolicy",
                "iot:AttachPrincipalPolicy",
                "iot:AttachThingPrincipal",
                "iot:CancelCertificateTransfer",
                "iot:CancelJob",
                "iot:CancelJobExecution",
                "iot:ClearDefaultAuthorizer",
                "iot:CreateAuthorizer",
                "iot:CreateCertificateFromCsr",
                "iot:CreateJob",
                "iot:CreateKeysAndCertificate",
                "iot:CreateOTAUpdate",
                "iot:CreatePolicy",
                "iot:CreatePolicyVersion",
                "iot:CreateRoleAlias",
```

```
                "iot:CreateStream",
                "iot:CreateThing",
                "iot:CreateThingGroup",
                "iot:CreateThingType",
                "iot:CreateTopicRule",
                "iot:DeleteAuthorizer",
                "iot:DeleteCACertificate",
                "iot:DeleteCertificate",
                "iot:DeleteJob",
                "iot:DeleteJobExecution",
                "iot:DeleteOTAUpdate",
                "iot:DeletePolicy",
                "iot:DeletePolicyVersion",
                "iot:DeleteRegistrationCode",
                "iot:DeleteRoleAlias",
                "iot:DeleteStream",
                "iot:DeleteThing",
                "iot:DeleteThingGroup",
                "iot:DeleteThingType",
                "iot:DeleteTopicRule",
                "iot:DeleteV2LoggingLevel",
                "iot:DeprecateThingType",
                "iot:DescribeAuthorizer",
                "iot:DescribeCACertificate",
                "iot:DescribeCertificate",
                "iot:DescribeDefaultAuthorizer",
                "iot:DescribeEndpoint",
                "iot:DescribeEventConfigurations",
                "iot:DescribeIndex",
                "iot:DescribeJob",
                "iot:DescribeJobExecution",
                "iot:DescribeRoleAlias",
                "iot:DescribeStream",
                "iot:DescribeThing",
                "iot:DescribeThingGroup",
                "iot:DescribeThingRegistrationTask",
                "iot:DescribeThingType",
                "iot:DetachPolicy",
                "iot:DetachPrincipalPolicy",
                "iot:DetachThingPrincipal",
                "iot:DisableTopicRule",
                "iot:EnableTopicRule",
                "iot:GetEffectivePolicies",
                "iot:GetIndexingConfiguration",
                "iot:GetJobDocument",
                "iot:GetLoggingOptions",
                "iot:GetOTAUpdate",
                "iot:GetPolicy",
                "iot:GetPolicyVersion",
                "iot:GetRegistrationCode",
                "iot:GetTopicRule",
                "iot:GetV2LoggingOptions",
                "iot:ListAttachedPolicies",
                "iot:ListAuthorizers",
                "iot:ListCACertificates",
                "iot:ListCertificates",
                "iot:ListCertificatesByCA",
                "iot:ListIndices",
                "iot:ListJobExecutionsForJob",
                "iot:ListJobExecutionsForThing",
                "iot:ListJobs",
                "iot:ListOTAUpdates",
                "iot:ListOutgoingCertificates",
                "iot:ListPolicies",
                "iot:ListPolicyPrincipals",
                "iot:ListPolicyVersions",
```

```
                "iot:ListPrincipalPolicies",
                "iot:ListPrincipalThings",
                "iot:ListRoleAliases",
                "iot:ListStreams",
                "iot:ListTargetsForPolicy",
                "iot:ListThingGroups",
                "iot:ListThingGroupsForThing",
                "iot:ListThingPrincipals",
                "iot:ListThingRegistrationTaskReports",
                "iot:ListThingRegistrationTasks",
                "iot:ListThings",
                "iot:ListThingsInThingGroup",
                "iot:ListThingTypes",
                "iot:ListTopicRules",
                "iot:ListV2LoggingLevels",
                "iot:RegisterCACertificate",
                "iot:RegisterCertificate",
                "iot:RegisterThing",
                "iot:RejectCertificateTransfer",
                "iot:RemoveThingFromThingGroup",
                "iot:ReplaceTopicRule",
                "iot:SearchIndex",
                "iot:SetDefaultAuthorizer",
                "iot:SetDefaultPolicyVersion",
                "iot:SetLoggingOptions",
                "iot:SetV2LoggingLevel",
                "iot:SetV2LoggingOptions",
                "iot:StartThingRegistrationTask",
                "iot:StopThingRegistrationTask",
                "iot:TestAuthorization",
                "iot:TestInvokeAuthorizer",
                "iot:TransferCertificate",
                "iot:UpdateAuthorizer",
                "iot:UpdateCACertificate",
                "iot:UpdateCertificate",
                "iot:UpdateEventConfigurations",
                "iot:UpdateIndexingConfiguration",
                "iot:UpdateRoleAlias",
                "iot:UpdateStream",
                "iot:UpdateThing",
                "iot:UpdateThingGroup",
                "iot:UpdateThingGroupsForThing",
                "iot:UpdateAccountAuditConfiguration",
                "iot:DescribeAccountAuditConfiguration",
                "iot:DeleteAccountAuditConfiguration",
                "iot:StartOnDemandAuditTask",
                "iot:CancelAuditTask",
                "iot:DescribeAuditTask",
                "iot:ListAuditTasks",
                "iot:CreateScheduledAudit",
                "iot:UpdateScheduledAudit",
                "iot:DeleteScheduledAudit",
                "iot:DescribeScheduledAudit",
                "iot:ListScheduledAudits",
                "iot:ListAuditFindings",
                "iot:CreateSecurityProfile",
                "iot:DescribeSecurityProfile",
                "iot:UpdateSecurityProfile",
                "iot:DeleteSecurityProfile",
                "iot:AttachSecurityProfile",
                "iot:DetachSecurityProfile",
                "iot:ListSecurityProfiles",
                "iot:ListSecurityProfilesForTarget",
                "iot:ListTargetsForSecurityProfile",
                "iot:ListActiveViolations",
                "iot:ListViolationEvents",
```

```
                "iot:ValidateSecurityProfileBehaviors"
            ],
            "Resource": "*"
        }
    ]
}
```

AWSpolítica gerenciada: AWSIoTConfigReadOnlyAccess

É possível anexar a política `AWSIoTConfigReadOnlyAccess` a suas identidades do IAM.

Essa política concede as permissões de identidade associadas que oferecem acesso somente leitura à todosAWS IoToperações de configuração. Para exibir esta política noAWS Management ConsoleConsulte,AWSIoTConfigReadOnlyAccess.

Detalhes da permissão

Esta política inclui as seguintes permissões.

- `iot`— Execute operações somente leitura das ações de configuração da IoT.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:DescribeAuthorizer",
                "iot:DescribeCACertificate",
                "iot:DescribeCertificate",
                "iot:DescribeDefaultAuthorizer",
                "iot:DescribeEndpoint",
                "iot:DescribeEventConfigurations",
                "iot:DescribeIndex",
                "iot:DescribeJob",
                "iot:DescribeJobExecution",
                "iot:DescribeRoleAlias",
                "iot:DescribeStream",
                "iot:DescribeThing",
                "iot:DescribeThingGroup",
                "iot:DescribeThingRegistrationTask",
                "iot:DescribeThingType",
                "iot:GetEffectivePolicies",
                "iot:GetIndexingConfiguration",
                "iot:GetJobDocument",
                "iot:GetLoggingOptions",
                "iot:GetOTAUpdate",
                "iot:GetPolicy",
                "iot:GetPolicyVersion",
                "iot:GetRegistrationCode",
                "iot:GetTopicRule",
                "iot:GetV2LoggingOptions",
                "iot:ListAttachedPolicies",
                "iot:ListAuthorizers",
                "iot:ListCACertificates",
```

```
                "iot:ListCertificates",
                "iot:ListCertificatesByCA",
                "iot:ListIndices",
                "iot:ListJobExecutionsForJob",
                "iot:ListJobExecutionsForThing",
                "iot:ListJobs",
                "iot:ListOTAUpdates",
                "iot:ListOutgoingCertificates",
                "iot:ListPolicies",
                "iot:ListPolicyPrincipals",
                "iot:ListPolicyVersions",
                "iot:ListPrincipalPolicies",
                "iot:ListPrincipalThings",
                "iot:ListRoleAliases",
                "iot:ListStreams",
                "iot:ListTargetsForPolicy",
                "iot:ListThingGroups",
                "iot:ListThingGroupsForThing",
                "iot:ListThingPrincipals",
                "iot:ListThingRegistrationTaskReports",
                "iot:ListThingRegistrationTasks",
                "iot:ListThings",
                "iot:ListThingsInThingGroup",
                "iot:ListThingTypes",
                "iot:ListTopicRules",
                "iot:ListV2LoggingLevels",
                "iot:SearchIndex",
                "iot:TestAuthorization",
                "iot:TestInvokeAuthorizer",
                "iot:DescribeAccountAuditConfiguration",
                "iot:DescribeAuditTask",
                "iot:ListAuditTasks",
                "iot:DescribeScheduledAudit",
                "iot:ListScheduledAudits",
                "iot:ListAuditFindings",
                "iot:DescribeSecurityProfile",
                "iot:ListSecurityProfiles",
                "iot:ListSecurityProfilesForTarget",
                "iot:ListTargetsForSecurityProfile",
                "iot:ListActiveViolations",
                "iot:ListViolationEvents",
                "iot:ValidateSecurityProfileBehaviors"
            ],
            "Resource": "*"
        }
    ]
}
```

AWSpolítica gerenciada: AWSIoTDataAccess

É possível anexar a política `AWSIoTDataAccess` a suas identidades do IAM.

Essa política concede permissões de identidade associadas que permitem acesso a todosAWS IoTOperações de dados. As operações de dados enviam dados por protocolos MQTT ou HTTP. Para exibir esta política noAWS Management ConsoleConsulte,`AWSIoTDataAccess`.

Detalhes da permissão

Esta política inclui as seguintes permissões.

- `iot`— RecuperarAWS IoTdados e permitir acesso total aoAWS IoTAções de mensagens.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:Connect",
                "iot:Publish",
                "iot:Subscribe",
                "iot:Receive",
                "iot:GetThingShadow",
                "iot:UpdateThingShadow",
                "iot:DeleteThingShadow",
                "iot:ListNamedShadowsForThing"
            ],
            "Resource": "*"
        }
    ]
}
```

AWSpolítica gerenciada: AWSIoTFullAccess

É possível anexar a política `AWSIoTFullAccess` a suas identidades do IAM.

Essa política concede permissões de identidade associadas que permitem acesso a todosAWS IoToperações de configuração e mensagens. Para exibir esta política noAWS Management ConsoleConsulte,`AWSIoTFullAccess`.

Detalhes da permissão

Esta política inclui as seguintes permissões.

- `iot`— RecuperarAWS IoTdados e permitir acesso total aoAWS IoTações de configuração e mensagens.
- `iotjobsdata`— RecuperarAWS IoTDados de trabalhos e permitir acesso total aAWS IoTOperações da API do plano de dados do trabalho

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

AWSpolítica gerenciada: AWSIoTLogging

É possível anexar a política `AWSIoTLogging` a suas identidades do IAM.

Esta política concede permissões de identidade associadas que permitem acesso para criar a Amazon CloudWatch Registra grupos e transmita logs para os grupos. Essa política está anexada à sua função de registro do CloudWatch. Para exibir esta política noAWS Management ConsoleConsulte,`AWSIoTLogging`.

Detalhes da permissão

Esta política inclui as seguintes permissões.

- `logs`— Recuperar CloudWatch logs. Também permite a criação de CloudWatch Registra grupos e transmita logs para os grupos.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogGroup",
                "logs:CreateLogStream",
                "logs:PutLogEvents",
                "logs:PutMetricFilter",
                "logs:PutRetentionPolicy",
                "logs:GetLogEvents",
                "logs:DeleteLogStream"
            ],
            "Resource": [
                "*"
            ]
        }
    ]
}
```

AWSpolítica gerenciada: AWSIoTOTAUpdate

É possível anexar a política `AWSIoTOTAUpdate` a suas identidades do IAM.

Essa política concede permissões de identidade associadas que permitem o acesso para criaçãoAWS IoTtrabalhos,AWS IoTtrabalhos de assinatura de código e para descreverAWStrabalhos de signatário de código. Para exibir esta política noAWS Management ConsoleConsulte,`AWSIoTOTAUpdate.`

Detalhes da permissão

Esta política inclui as seguintes permissões.

- `iot`— CriarAWS IoTTrabalhos e trabalhos de assinatura de código.

- `signer`— Execute a criação deAWStrabalhos de signatário de código.

```
{
    "Version": "2012-10-17",
    "Statement": {
        "Effect": "Allow",
        "Action": [
            "iot:CreateJob",
            "signer:DescribeSigningJob"
        ],
        "Resource": "*"
    }
}
```

AWSpolítica gerenciada: AWSIoTRuleActions

É possível anexar a política `AWSIoTRuleActions` a suas identidades do IAM.

Essa política concede permissões de identidade associadas que permitem acesso a todosAWS service (Serviço da AWS)s compatíveis com oAWS IoTAções de regra do. Para exibir esta política noAWS Management ConsoleConsulte,`AWSIoTRuleActions`.

Detalhes da permissão

Esta política inclui as seguintes permissões.

- `iot`- Execute ações para publicar mensagens de ação de regra.
- `dynamodb`- Insira uma mensagem em uma tabela do DynamoDB ou divida uma mensagem em várias colunas de uma tabela do DynamoDB.
- `s3`- Armazene um objeto em um bucket do Amazon S3.
- `kinesis`- Envie uma mensagem para um objeto de fluxo do Amazon Kinesis.
- `firehose`- Insira um registro em um objeto de fluxo do Kinesis Data Firehose.
- `cloudwatch`- Altere CloudWatch estado de alarme ou enviar dados de mensagem para CloudWatch Métrica do.
- `sns`- Execute a operação para publicar uma notificação usando o Amazon SNS. Esta operação tem escopo paraAWS IoTTópicos do SNS.
- `sqs`- Insira uma mensagem para adicionar à fila SQS.
- `es`- Envie uma mensagem para o OpenSearch Serviço de serviço.

```
{
    "Version": "2012-10-17",
    "Statement": {
        "Effect": "Allow",
        "Action": [
            "dynamodb:PutItem",
            "kinesis:PutRecord",
            "iot:Publish",
            "s3:PutObject",
            "sns:Publish",
```

```
            "sqs:SendMessage*",
            "cloudwatch:SetAlarmState",
            "cloudwatch:PutMetricData",
            "es:ESHttpPut",
            "firehose:PutRecord"
        ],
        "Resource": "*"
    }
}
```

AWSpolítica gerenciada: AWSIoTThingsRegistration

É possível anexar a política `AWSIoTThingsRegistration` a suas identidades do IAM.

Essa política concede às permissões de identidade associadas que permitem acesso para registrar coisas em massa usando o`StartThingRegistrationTask`API. Essa política pode afetar o processamento de dados e armazenamento. Para exibir esta política noAWS Management ConsoleConsulte,`AWSIoTThingsRegistration`.

Detalhes da permissão

Esta política inclui as seguintes permissões.

- `iot`- Execute ações para criar coisas e anexar políticas e certificados ao se registrar em massa.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:AddThingToThingGroup",
                "iot:AttachPolicy",
                "iot:AttachPrincipalPolicy",
                "iot:AttachThingPrincipal",
                "iot:CreateCertificateFromCsr",
                "iot:CreatePolicy",
                "iot:CreateThing",
                "iot:DescribeCertificate",
                "iot:DescribeThing",
                "iot:DescribeThingGroup",
                "iot:DescribeThingType",
                "iot:DetachPolicy",
                "iot:DetachThingPrincipal",
                "iot:GetPolicy",
                "iot:ListAttachedPolicies",
                "iot:ListPolicyPrincipals",
                "iot:ListPrincipalPolicies",
                "iot:ListPrincipalThings",
                "iot:ListTargetsForPolicy",
                "iot:ListThingGroupsForThing",
                "iot:ListThingPrincipals",
                "iot:RegisterCertificate",
                "iot:RegisterThing",
                "iot:RemoveThingFromThingGroup",
                "iot:UpdateCertificate",
```

```
                "iot:UpdateThing",
                "iot:UpdateThingGroupsForThing",
                "iot:AddThingToBillingGroup",
                "iot:DescribeBillingGroup",
                "iot:RemoveThingFromBillingGroup"
            ],
            "Resource": [
                "*"
            ]
        }
    ]
}
```

Atualizações do AWS IoT para políticas gerenciadas pela AWS

Visualizar detalhes sobre atualizações em políticas gerenciadas pela AWS para o AWS IoT desde que esse serviço começou a rastrear essas alterações. Para receber alertas automáticos sobre alterações feitas nesta página, inscreva-se no feed RSS na página de histórico de documentos do AWS IoT.

Alteração	Descrição	Data
AWSIoTFullAccess (p. 413)— Atualizar para uma política existente	AWS IoTadicionou novas permissões para permitir que os usuários acessem oAWS IoTOperações de API de plano de dados de tarefas usando o protocolo HTTP. Um novo prefixo de política do IAM`iotjobsdata:`, fornece controle de acesso mais refinado para acessarAWS IoTPontos finais do plano de dados de trabalhos. Para operações de API de plano de controle, você ainda usa `oiot:`prefixo. Para obter mais informações, consulteAWS IoT CorePolíticas para protocolo HTTPS (p. 773)	11 de maio de 2022
O AWS IoT iniciou o rastreamento das alterações	O AWS IoT começou a monitorar as alterações para as políticas gerenciadas da AWS.	11 de maio de 2022

Solução de problemas de identidade e acesso do AWS IoT

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

Tópicos

- Não tenho autorização para executar uma ação no AWS IoT (p. 418)

- Não estou autorizado a executar iam:PassRole (p. 418)
- Quero visualizar minhas chaves de acesso (p. 419)
- Sou administrador e desejo permitir que outras pessoas tenham acesso ao AWS IoT (p. 419)
- Quero permitir que as pessoas fora da minha Conta da AWS acessem meus recursos do AWS IoT (p. 419)

Não tenho autorização para executar uma ação no AWS IoT

Se o AWS Management Console informar que você não está autorizado a executar uma ação, você deverá entrar em contato com o administrador para obter assistência. O administrador é a pessoa que forneceu a você o seu nome de usuário e senha.

O exemplo a seguir ocorre quando `omateojackson`O usuário do IAM tenta usar o console para visualizar detalhes sobre uma coisa, mas não tem`iot:`*`DescribeThing`*permissões.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 iot:DescribeThing
            on resource: MyIoTThing
```

Neste caso, Mateo pede ao administrador para atualizar suas políticas para permitir a ele o acesso ao recurso *`MyIoTThing`* usando a ação `iot:`*`DescribeThing`*.

O uso doAWS IoTDevice Advisor

Se você estiver usandoAWS IoTDevice Advisor, o exemplo a seguir de erro ocorre quando `omateojackson`O usuário do IAM tenta usar o console para visualizar detalhes sobre uma definição de suíte, mas não tem`iotdeviceadvisor:GetSuiteDefinition`permissões.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 iotdeviceadvisor:GetSuiteDefinition
            on resource: MySuiteDefinition
```

Neste caso, Mateo pede ao administrador para atualizar suas políticas para permitir que ele acesse o*`MySuiteDefinition`*recurso usando `o`iotdeviceadvisor:*`GetSuiteDefinition`*Ação .

Não estou autorizado a executar iam:PassRole

Se você receber uma mensagem de erro informando que você não está autorizado a executar a ação `iam:PassRole`, entre em contato com o administrador para obter assistência. O administrador é a pessoa que forneceu a você o seu nome de usuário e senha. Peça a essa pessoa para atualizar suas políticas para permitir que você passe uma função para o AWS IoT.

Alguns Serviços da AWS permitem que você transmita uma função existente para o serviço, em vez de criar uma função de serviço ou uma função vinculada ao serviço. Para fazer isso, um usuário deve ter permissões para passar o perfil para o serviço.

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

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

Neste caso, Mary pede ao administrador para atualizar suas políticas para permitir que ela execute a ação `iam:PassRole`.

Quero visualizar minhas chaves de acesso

Depois de criar suas chaves de acesso de usuário do IAM, é possível visualizar seu ID da chave de acesso a qualquer momento. No entanto, você não pode visualizar sua chave de acesso secreta novamente. Se você perder sua chave secreta, crie um novo par de chaves de acesso.

As chaves de acesso consistem em duas partes: um ID de chave de acesso (por exemplo, `AKIAIOSFODNN7EXAMPLE`) e uma chave de acesso secreta (por exemplo, `wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY`). Como um nome de usuário e uma senha, você deve usar o ID da chave de acesso e a chave de acesso secreta em conjunto para autenticar suas solicitações. Gerencie suas chaves de acesso de forma tão segura quanto você gerencia seu nome de usuário e sua senha.

> **Important**
>
> Não forneça as chaves de acesso a terceiros, mesmo que seja para ajudar a encontrar o ID de usuário canônico. Ao fazer isso, você pode dar a alguém acesso permanente à sua conta.

Ao criar um par de chaves de acesso, você é solicitado a guardar o ID da chave de acesso e a chave de acesso secreta em um local seguro. A chave de acesso secreta só está disponível no momento em que é criada. Se você perder sua chave de acesso secreta, será necessário adicionar novas chaves de acesso para seu usuário do IAM. Você pode ter no máximo duas chaves de acesso. Se você já tiver duas, você deverá excluir um par de chaves para poder criar um novo. Para visualizar as instruções, consulte Gerenciar chaves de acesso no Guia do usuário do IAM.

Sou administrador e desejo permitir que outras pessoas tenham acesso ao AWS IoT

Para permitir que outros usuários acessem o AWS IoT, crie uma entidade do IAM (usuário ou função) para a pessoa ou a aplicação que precisa do acesso. Elas usarão as credenciais dessa entidade para acessar a AWS. Você deve anexar uma política à entidade que concede a eles as permissões corretas no AWS IoT.

Para começar a usar imediatamente, consulte Criar os primeiros usuário e grupo delegados do IAM no Guia do usuário do IAM.

Quero permitir que as pessoas fora da minha Conta da AWS acessem meus recursos do AWS IoT

Você pode criar uma função que os usuários de outras contas ou pessoas fora da sua organização podem usar para acessar seus recursos. Você pode especificar quem é confiável para assumir a função. 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 o seguinte:

- Para saber se o AWS IoT oferece suporte a esses recursos, consulte Como o AWS IoT funciona com o IAM (p. 388).
- Para saber como conceder acesso a seus recursos em todas as Contas da AWS pertencentes a você, consulte Fornecimento de acesso a um usuário do IAM em outra Conta da AWS pertencente a você no Guia de usuário do IAM.
- Para saber como conceder acesso a seus recursos para terceiros Contas da AWS, consulte Fornecimento de acesso a Contas da AWS pertencentes a terceiros no Guia do usuário do IAM.
- Para saber como conceder acesso por meio da federação de identidades, consulte Conceder acesso a usuários autenticados externamente (federação de identidades) no Guia do usuário do IAM.
- Para saber a diferença entre usar funções e políticas baseadas em recursos para acesso entre contas, consulte Como as funções do IAM diferem de políticas baseadas em recursos no Guia do usuário do IAM.

Registro em log e monitoramento

O monitoramento é uma parte importante da manutenção da confiabilidade, da disponibilidade e da performance do AWS IoT e de suas soluções da AWS. Você deve coletar dados de monitoramento de todas as partes de sua solução da AWS para facilitar a depuração de uma falha multipontos, caso ela ocorra. Para obter informações sobre procedimentos de registro em log e monitoramento, consulte Monitorar o AWS IoT (p. 432)

Ferramentas de monitoramento

A AWS fornece ferramentas que você pode usar para monitorar o AWS IoT. 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:

- Amazônia CloudWatch Alarmes— observe uma única métrica ao longo de um período que você especificar e realize uma ou mais ações com base no valor da métrica em relação a um determinado limite ao longo de vários períodos. A ação é uma notificação enviada para um tópico do Amazon Simple Notification Service (Amazon SNS) ou uma política do Amazon EC2 Auto Scaling. CloudWatch Os alarmes do não invocam ações só porque estão em um determinado estado. O estado deve ter sido alterado e mantido por uma quantidade especificada de períodos. Para obter mais informações, consulteMonitorAWS IoTalarmes e métricas usando o Amazon CloudWatch (p. 439)
- Amazônia CloudWatch Logs— Monitore, armazene e acesse seus arquivos de log doAWS CloudTrailou outras fontes. Amazônia CloudWatch Os Logs também permitem que você veja etapas críticasAWS IoTOs casos de teste do Device Advisor recebem, geraram eventos e mensagens MQTT enviadas de seus dispositivos ouAWS IoT Coredurante a execução do teste. Esses registros permitem depurar e executar ações corretivas em seus dispositivos. Para obter mais informações, consulteMonitorAWS IoTusando CloudWatch Logs (p. 454)Para obter mais informações sobre como usar o Amazon CloudWatch, consulteMonitorar arquivos de lognoAmazônia CloudWatch Guia do usuário do.
- Amazônia CloudWatch Eventos— faça correspondência de eventos e direcione-os a uma ou mais funções ou fluxos de destino para fazer alterações, capturar informações de estado e realizar ações corretivas. Para obter mais informações, consulteO que é a Amazon CloudWatch EventosnoAmazônia CloudWatch Guia do usuário do.
- AWS CloudTrailMonitorar logs do— Compartilhe arquivos de log entre contas, monitore CloudTrail Registre arquivos em tempo real enviando-os para CloudWatch Logs, escreva aplicativos 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, consulteRegistre em log as chamadas à API do AWS IoT usando o AWS CloudTrail. (p. 475)e tambémTrabalho com CloudTrail Arquivos de lognoAWS CloudTrailGuia do usuário do.

Ferramentas de monitoramento manual

Outra parte importante do monitoramentoAWS IoTenvolve monitorar manualmente esses itens que o CloudWatch alarmes não cobrem. OAWS IoT, CloudWatch e outrosAWSpainéis de console de serviço fornecem um at-a-glance Visualização do estado doAWSEnvironment. Recomendamos que você também verifique os arquivos de registro do AWS IoT.

- O painel da AWS IoT mostra:
  - Certificados de CA

  - Certificados
  - Políticas
  - Regras
  - Coisas
- A página inicial do CloudWatch mostra:
  - Alarmes e status atual.
  - Gráficos de alarmes e recursos.
  - Estado de integridade do serviço.

  Você pode usar CloudWatch Para fazer o seguinte:
  - Criar painéis personalizados para monitorar os serviços com os quais você se preocupa.
  - Colocar em gráfico dados de métrica para solucionar problemas e descobrir tendências.
  - Pesquise e procure todas as métricas de recursos da AWS.
  - Criar e editar alertas para ser notificado sobre problemas.

Validação de conformidade doAWS IoTServiços

Auditores externos avaliam a segurança e a conformidade dos serviços da Serviços da AWS como parte de vários programas de conformidade da AWS, como SOC, PCI, FedRAMP e HIPAA.

Para saber se o AWS IoT ou outros Serviços da AWS estão no escopo de programas específicos de conformidade, consulte Serviços da AWS no escopo por programa de conformidade. Para obter informações gerais, consulte Programas de conformidade da AWS.

É possível fazer download de relatórios de auditoria de terceiros usando o AWS Artifact. Para obter mais informações, consulte Downloading Reports in AWS Artifact.

Sua responsabilidade de conformidade ao usar o Serviços da AWS é determinada pela confidencialidade dos seus dados, pelos objetivos de conformidade da sua empresa e pelos regulamentos e leis aplicáveis. A AWS fornece os seguintes recursos para ajudar com a conformidade:

- Guias de referência rápida de conformidade e segurança: esses guias de implantação abordam as considerações de arquitetura e fornecem etapas para implantação de ambientes de lista de referência na AWS concentrados em conformidade e segurança.
- Whitepaper Elaboração de arquitetura para segurança e conformidade com HIPAA: esse whitepaper descreve como as empresas podem usar a AWS para criar aplicações adequadas aos padrões HIPAA.

  > **Note**
  >
  > Nem todos os Serviços da AWS estão qualificados pela HIPAA. Para mais informações, consulte a Referência dos serviços qualificados pela HIPAA.
- Recursos de conformidade da AWS: essa coleção de manuais e guias pode ser aplicada a seu setor e local.
- Avaliar recursos com regras no AWS Config Developer Guide (Guia do desenvolvedor do CCI): o serviço AWS Config avalia como as configurações de recursos estão em conformidade com práticas internas, diretrizes do setor e regulamentos.
- AWS Security Hub: esse AWS service (Serviço da AWS) fornece uma visão abrangente do estado de sua segurança na AWS que ajuda você a conferir sua conformidade com padrões e práticas recomendadas de segurança do setor.
- AWS Audit Manager: esse AWS service (Serviço da AWS) ajuda a auditar continuamente seu uso da AWS para simplificar a forma como você gerencia os riscos e a conformidade com regulamentos e padrões do setor.

Resiliência noAWSIoT Core

A infraestrutura global da AWS se baseia em Região da AWS e zonas de disponibilidade. A Região da AWS oferece várias zonas de disponibilidade separadas e isoladas fisicamente que são conectadas com baixa latência, altas taxas de transferência e em 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 data center tradicionais.

Para obter mais informações sobreRegião da AWSs e zonas de disponibilidade, consulteAWSInfraestrutura global.

O AWS IoT Core armazena informações sobre seus dispositivos no registro do dispositivo. Ele também armazena certificados da CA, certificados do dispositivo e dados de sombra do dispositivo. Quando houver falhas de hardware ou de rede, esses dados são replicados automaticamente em zonas de disponibilidade, mas não em regiões.

AWS IoT CoreO publica eventos MQTT quando o registro do dispositivo é atualizado. É possível usar essas mensagens para fazer backup de seus dados de registro e salvá-los em outro lugar, como uma tabela do DynamoDB. Você é responsável por salvar os certificados que o AWS IoT Core cria para você ou aqueles que você mesmo cria. A sombra do dispositivo armazena dados de estado sobre os seus dispositivos e pode ser reenviada quando um dispositivo voltar a ficar online.AWS IoT O Device Advisor armazena informações sobre a configuração do conjunto de testes. Esses dados são replicados automaticamente no caso de falhas de hardware ou de rede.

AWS IoT CoreOs recursos do são específicos da região e não são replicados emRegiões da AWSa menos que você faça isso especificamente.

Para obter informações sobre as melhores práticas de segurança, consulteMelhores práticas de segurança no AWS IoT Core (p. 425).

Usar o AWS IoT Core com endpoints da VPC de interface

comAWS IoT Core, você pode criarEndpoints de dados de IoTdentro de sua VPC usandoVPC endpoints de interface. Os endpoints de interface do são desenvolvidos peloAWSPrivateLink, umAWSTecnologia que você pode usar para acessar os serviços em execução noAWSusando endereços IP privados. Para obter mais informações, consulte o Amazon Virtual Private Cloud.

Para conectar dispositivos em campo em redes remotas, como uma rede corporativa à suaAWSVPC, consulte as várias opções listadas naMatriz de conectividade entre a rede e a Amazon VPC.

> **Note**
>
> Os endpoints de VPC para IoT Core não são suportados no momentoAWSRegiões da China.

Tópicos do capítulo:

- Criar VPC endpoints para o AWS IoT Core (p. 423)
- Controle do acesso aoAWS IoT CoreVPC endpoints (p. 423)
- Limitações dos endpoints da VPC (p. 424)
- Dimensionando endpoints da VPC com IoT Core (p. 424)
- Uso de domínios personalizados com VPC endpoints (p. 424)
- Disponibilidade de VPC endpoints paraAWS IoT Core (p. 424)

Criar VPC endpoints para o AWS IoT Core

Para começar a usar endpoints da VPC, bastaCriar um VPC endpoint de interfacee selecioneAWS IoT Corecomo oAWSserviçoServiço. Se você estiver usando a CLI, primeiro ligue*describe-vpc-endpoint-services*para garantir que você esteja escolhendo uma zona de disponibilidade ondeAWS IoT Coreestá presente em seu particularRegião da AWS. Por exemplo, em us-east-1, esse comando terá a seguinte aparência:

```
aws ec2 describe-vpc-endpoint-services --service-name com.amazonaws.us-east-1.iot.data
```

> **Note**
>
> O recurso VPC para criar automaticamente um registro DNS está desativado. Para se conectar a esses endpoints, você deve criar manualmente um registro DNS privado. Para obter mais informações sobre registros DNS VPC privada, consulteDNS privado para endpoints de interface. Para obter mais informações sobreAWS IoT CoreLimitações de VPC, consulteLimitações dos endpoints da VPC (p. 424).

Para conectar clientes MQTT às interfaces de endpoint da VPC, você deve criar manualmente registros DNS em uma Zona Hospedada Privada anexada à sua VPC. Para começar a usar, consulteCriar uma zona hospedada privada. Dentro da sua Zona Hospedada Privada, crie um registro de alias para cada IP de elastic network interface para o endpoint da VPC. Se você tiver vários IPs de interface de rede para vários endpoints de VPC, crie registros DNS ponderados com pesos iguais em todos os registros ponderados. Esses endereços IP estão disponíveis no*DescribeNetworkInterfaces*Chamada de API quando filtrada pelo ID do endpoint da VPC no campo de descrição.

Controle do acesso aoAWS IoT CoreVPC endpoints

Você pode restringir o acesso do dispositivo aoAWS IoT Corepara ser permitido somente por meio do VPC endpoint usando VPCChaves de contexto de condição.AWS IoT Coresuporta as seguintes chaves de contexto relacionadas à VPC:

- SourceVpc
- SourceVpce
- VpcSourceIp

> **Note**
>
> AWS IoT CoreO não oferece suporte aohttps://docs.aws.amazon.com/vpc/latest/privatelink/vpc-endpoints-access.html#vpc-endpoint-policiesPolíticas de endpoint de VPC no momento.

Por exemplo, a política a seguir concede permissão para se conectar aoAWS IoT CoreUsando um ID de cliente que corresponda ao nome da coisa e para publicar em qualquer tópico prefixado pelo nome da coisa, condicional no dispositivo se conectar a um endpoint de VPC com um ID de endpoint de VPC específico. Essa política negaria tentativas de conexão com seu endpoint de dados de IoT público.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:Connect"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:client/
${iot:Connection.Thing.ThingName}"
```

```
            ],
            "Condition": {
                "StringEquals": {
                    "aws:SourceVpce": "vpce-1a2b3c4d"
                }
            }

        },
        {
            "Effect": "Allow",
            "Action": [
                "iot:Publish"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:topic/
${iot:Connection.Thing.ThingName}/*"
            ]
        }
    ]
}
```

Limitações dos endpoints da VPC

Esta seção abrange as limitações dos VPC endpoints em comparação com endpoints públicos.

- No momento, VPC endpoints são compatíveis com oEndpoints de dados de IoTapenas
- Os períodos de manutenção da MQTT são limitados a 230 segundos. Manter alives por mais tempo que esse período será automaticamente reduzido para 230 segundos
- Cada endpoint VPC oferece suporte a 100.000 dispositivos conectados simultâneos no total. Se você precisar de mais conexões, consulteDimensionando endpoints da VPC com IoT Core (p. 424).
- Os VPC endpoints são compatíveis somente com tráfego IPv4.
- VPC endpoints servirãoCertificados ATSsomente, exceto para domínios personalizados.
- Políticas de VPC endpointNo momento, não são suportados.
- Para endpoints da VPC que são criados para oAWS IoT Coreplano de dados,AWS IoT Corenão oferece suporte ao uso de registros DNS públicos zonais ou regionais.

Dimensionando endpoints da VPC com IoT Core

AWS IoT CoreOs endpoints de VPC de interface são limitados a 100.000 dispositivos conectados em um único endpoint de interface. Se o seu caso de uso exigir mais conexões simultâneas com o broker, recomendamos usar vários endpoints de VPC e rotear manualmente seus dispositivos pelos endpoints da interface. Ao criar registros DNS privados para rotear o tráfego para seus endpoints da VPC, certifique-se de criar tantos registros ponderados quanto você tiver endpoints de VPC para distribuir o tráfego em vários endpoints.

Uso de domínios personalizados com VPC endpoints

Se você quiser usar domínios personalizados com endpoints da VPC, você deve criar seus registros de nome de domínio personalizados em uma Zona hospedada privada e criar registros de roteamento na Route53. Para obter mais informações, consulteCriar uma zona hospedada privada.

Disponibilidade de VPC endpoints paraAWS IoT Core

AWS IoT CoreOs endpoints de VPC de interface estão disponíveis em todosAWS IoT CoreRegiões compatíveis, com exceção deAWSRegiões da China.

Segurança da infraestrutura no AWS IoT

Como uma coleção de serviços gerenciados,AWS IoTé protegido peloAWSProcedimentos de segurança de rede global descritos naAmazon Web Services: Visão geral dos processos de segurançaWhitepaper.

Você usa chamadas de API publicadas pela AWS para acessar o AWS IoT por meio da rede. Os clientes devem oferecer suporte a Transport Layer Security (TLS) 1.2 ou posterior. Os clientes também devem ter suporte a conjuntos de criptografia com perfect forward secrecy (PFS) como Ephemeral Diffie-Hellman (DHE) ou Ephemeral Elliptic Curve Diffie-Hellman (ECDHE). A maioria dos sistemas modernos, como o Java 7 e versões posteriores, oferece suporte a esses modos. Para obter mais informações, consulteSegurança de transporte no AWS IoT (p. 381)

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

Monitoramento de segurança de frotas de produção ou dispositivos comAWS IoTServiços

Frotas de IoT consistem em grandes quantidades de dispositivos com capacidades diversas, duradouros e geograficamente distribuídos. Essas características tornam a configuração da frota complexa epropenso a erros. E como os dispositivos, quase sempre, têm restrições quanto à capacidade computacional e aos recursos de memória e armazenamento, isso limita o uso de criptografia e outras formas de segurança nos próprios dispositivos. Além disso, muitas vezes, os dispositivos usam software com vulnerabilidades conhecidas. Esses fatores tornam frotas de IoT um alvo atrativo para hackers e tornam difícil proteger a frota de dispositivos de forma contínua.

O AWS IoT Device Defender aborda esses desafios fornecendo ferramentas para identificar problemas de segurança e desvios das melhores práticas. Você pode usarAWS IoTDevice Defender para analisar, auditar e monitorar dispositivos conectados para detectar comportamentos anormais e reduzir os riscos de segurança.AWS IoT Device DefenderO pode auditar frotas de dispositivos para garantir que elas cumpram as melhores práticas de segurança e detectem comportamentos anormais em dispositivos. Isso possibilita a aplicação de políticas de segurança consistentes em toda a sua frota de dispositivos do AWS IoT e responder rapidamente quando os dispositivos estiverem comprometidos. Para obter mais informações, consulteAWS IoT Device Defender (p. 882)

AWS IoTO Device Advisor envia atualizações e corrige sua frota conforme necessário.AWS IoT O Device Advisor atualiza casos de teste automaticamente. Os casos de testequeVocê selecionaestão sempre comVersão mais recente. Para obter mais informações, consulte Device Advisor (p. 1067).

Melhores práticas de segurança no AWS IoT Core

Esta seção contém informações sobre as melhores práticas de segurança para o AWS IoT Core. Para obter mais informações, consulte Dez regras de ouro de segurança para soluções de IoT.

Proteger conexões MQTT no AWS IoT

O AWS IoT Core é um serviço de nuvem gerenciado que permite que dispositivos conectados interajam com aplicativos em nuvem e outros dispositivos de forma fácil e segura. O AWS IoT Core oferece suporte a HTTP, WebSocket e MQTT, um protocolo de comunicação leve projetado especificamente para tolerar conexões intermitentes. Se você estiver se conectando aoAWS IoTAo usar MQTT, cada uma de suas conexões deve ser associada a um identificador conhecido como um ID de cliente. Os IDs de cliente

MQTT identificam exclusivamente conexões MQTT. Se uma nova conexão for estabelecida usando um ID de cliente que já está reivindicado para outra conexão, o agente de mensagens do AWS IoT cancelará a conexão antiga para permitir a nova conexão. Os IDs de cliente devem ser exclusivos dentro de cadaConta da AWSe cadaRegião da AWS. Isso significa que você não precisará impor a exclusividade global de IDs de cliente fora da suaConta da AWSou em todas as regiões dentro da suaConta da AWS.

O impacto e a gravidade de descartar conexões MQTT em sua frota de dispositivos dependem de vários fatores. Dentre elas estão:

- O caso de uso (por exemplo, os dados que os dispositivos enviam para a AWS IoT, a quantidade de dados e a frequência em que os dados são enviados).
- A configuração do cliente MQTT (por exemplo, as configurações de reconexão automática, as temporizações de retirada associadas e o uso de Sessões MQTT persistentes (p. 88)).
- Restrições de recursos de dispositivo.
- A causa raiz das desconexões, sua agressividade e persistência.

Para evitar conflitos de ID de cliente e os possíveis impactos negativos, certifique-se de que cada dispositivo ou aplicativo móvel tenha umAWS IoTou política do IAM que restringe quais IDs de cliente podem ser usadas para conexões MQTT com oAWS IoTAgente de mensagens. Por exemplo, você pode usar uma política do IAM para impedir que um dispositivo feche involuntariamente a conexão de outro dispositivo usando um ID de cliente que já esteja em uso. Para obter mais informações, consulteAutorização (p. 330)

Todos os dispositivos em sua frota devem ter credenciais com privilégios que autorizem apenas ações planejadas, incluindo, mas não se limitando a, ações MQTT do AWS IoT, como publicar mensagens ou assinar tópicos com escopo e contexto específicos. As políticas de permissão específicas podem variar para seus casos de uso. Identifique as políticas de permissão que melhor atendem aos seus requisitos de negócios e de segurança.

Para simplificar a criação e o gerenciamento de políticas de permissão, você pode usarVariáveis de política do AWS IoT Core (p. 336)eVariáveis de política do IAM. As variáveis de políticas podem ser colocadas em uma política e, quando a política for avaliada, as variáveis serão substituídas por valores fornecidos na solicitação do dispositivo. Usando variáveis de políticas, você pode criar uma única política para conceder permissões a vários dispositivos. Você pode identificar as variáveis de políticas relevantes para seu caso de uso com base na configuração de sua conta do AWS IoT, no mecanismo de autenticação e no protocolo de rede usado na conexão ao agente de mensagens do AWS IoT. No entanto, para escrever as melhores políticas de permissão, você precisa considerar informações específicas sobre seu caso de uso e o modelo de ameaça.

Por exemplo, se você registrou seus dispositivos noAWS IoTregistro, você pode usarVariáveis de política de coisas (p. 338)emAWS IoTPolíticas para conceder ou negar permissões com base em propriedades de coisas, como nomes de coisas, tipos de coisa e valores de atributo de coisas. O nome da coisa é obtido do ID do cliente na mensagem de conexão MQTT enviada quando uma coisa se conecta ao AWS IoT. A coisa que as variáveis de política são substituídas quando uma coisa se conecta aAWS IoTsobre MQTT usando autenticação mútua TLS ou MQTT sobre o WebSocket protocolo usando autenticadoIdentidades do Amazon Cognito. Você pode usar oAttachThingPrincipalAPI para anexar certificados e identidades do Amazon Cognito autenticadas a uma coisa.`iot:Connection.Thing.ThingName`É uma variável de política de coisas útil para impor restrições de ID de cliente. A política de exemplo do AWS IoT a seguir exige que um nome de coisa registrado seja usado como o ID de cliente para conexões MQTT com o agente de mensagens do AWS IoT:

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

```
        "Resource":[
            "arn:aws:iot:us-east-1:123456789012:client/${iot:Connection.Thing.ThingName}"
        ]
    }
    ]
}
```

Se você deseja identificar conflitos de ID de cliente em andamento, você pode habilitar e usarCloudWatch Logs paraAWS IoT (p. 454). Para cada conexão MQTT da qual o agente de mensagens do AWS IoT se desconecta devido a conflitos de ID de cliente, um registro em log semelhante ao seguinte é gerado:

```
{
    "timestamp": "2019-04-28 22:05:30.105",
    "logLevel": "ERROR",
    "traceId": "02a04a93-0b3a-b608-a27c-1ae8ebdb032a",
    "accountId": "123456789012",
    "status": "Failure",
    "eventType": "Disconnect",
    "protocol": "MQTT",
    "clientId": "clientId01",
    "principalId": "1670fcf6de55adc1930169142405c4a2493d9eb5487127cd0091ca0193a3d3f6",
    "sourceIp": "203.0.113.1",
    "sourcePort": 21335,
    "reason": "DUPLICATE_CLIENT_ID",
    "details": "A new connection was established with the same client ID"
}
```

Você pode usar umFiltro CloudWatch Logscomo`{$.reason= "DUPLICATE_CLIENT_ID" }`para procurar instâncias de conflitos de ID do cliente ou para configurarFiltros de métrica CloudWatche correspondente CloudWatch alarmes para monitoramento e emissão de relatórios contínuos.

Você pode usarAWS IoTDevice Defenderpara identificar excessivamente permissivoAWS IoTe políticas do IAM.AWS IoT O Device Defender também fornece uma verificação de auditoria que notificará você se vários dispositivos em sua frota estiverem se conectando aoAWS IoTagente de mensagens usando a mesma ID do cliente.

Você pode usarAWS IoTDevice Advisor para validar que seus dispositivos podem se conectar de forma confiável aoAWS IoT Coree siga as melhores práticas de segurança do.

Consulte também

- AWS IoT Core
- Recursos de segurança do AWS IoT (p. 294)
- Variáveis de política do AWS IoT Core (p. 336)
- Variáveis de política do IAM
- Amazon Cognito Identity
- AWS IoT Device Defender
- CloudWatch Logs paraAWS IoT (p. 454)

Manter o relógio do dispositivo sincronizado

É importante ter a hora exata no seu dispositivo. Os certificados X.509 têm data e hora de expiração. O relógio em seu dispositivo é usado para verificar se um certificado de servidor ainda é válido. Se você estiver criando dispositivos comerciais de IoT, lembre-se de que seus produtos podem ser armazenados por períodos prolongados antes de serem vendidos. Os relógios em tempo real podem ter desvios durante

esse período, e as baterias podem ser descarregadas, portanto, definir a hora na definição de fábrica não é suficiente.

Para a maioria dos sistemas, isso significa que o software do dispositivo deve incluir um cliente de protocolo de tempo de rede (NTP). O dispositivo deve aguardar até ser sincronizado com um servidor NTP antes de tentar se conectar ao AWS IoT Core. Se isso não for possível, o sistema deve fornecer uma maneira de um usuário definir a hora do dispositivo para que as conexões subsequentes tenham êxito.

Depois que o dispositivo for sincronizado com um servidor NTP, ele poderá abrir uma conexão com o AWS IoT Core. A inclinação do relógio que é permitida depende do que você está tentando fazer com a conexão.

Validar o certificado do servidor

A primeira coisa que um dispositivo faz para interagir com o AWS IoT é abrir uma conexão segura. Ao conectar o dispositivo à AWS IoT, verifique se está falando com a AWS IoT e não com outro servidor se passando pela AWS IoT. Cada um dos servidores do AWS IoT é provisionado com um certificado emitido para o domínio `iot.amazonaws.com`. Este certificado foi emitido para o AWS IoT por uma autoridade de certificação confiável que verificou nossa identidade e propriedade do domínio.

Uma das primeiras coisas que o AWS IoT Core faz quando um dispositivo se conecta é enviar um certificado do servidor ao dispositivo. Os dispositivos podem verificar se eles esperavam se conectar ao `iot.amazonaws.com` e se o servidor no destino dessa conexão possui um certificado de uma autoridade confiável para esse domínio.

Os certificados TLS estão no formato X.509 e incluem uma grande variedade de informações, como nome, localização, nome de domínio e um período de validade da organização. O período de validade é especificado como um par de valores de tempo chamados `notBefore` e `notAfter`. Serviços como o AWS IoT Core usam períodos de validade limitados (por exemplo, um ano) para seus certificados de servidor e começam a atender outros novos antes de os antigos expirarem.

Usar uma identidade única por dispositivo

Use uma única identidade por cliente. Os dispositivos geralmente usam certificados de cliente X.509. Aplicativos web e móveis usam o Amazon Cognito Identity. Isso permite aplicar permissões refinadas aos seus dispositivos.

Por exemplo, você tem um aplicativo que consiste em um dispositivo móvel que recebe atualizações de status de dois objetos domésticos inteligentes diferentes: uma lâmpada e um termostato. A lâmpada envia o status do nível de bateria e um termostato envia mensagens que relatam a temperatura.

A AWS IoT autentica dispositivos individualmente e trata cada conexão individualmente. Você pode aplicar controles de acesso refinados usando políticas de autorização. É possível definir uma política para o termostato que permite que ele publique em um espaço de tópico. É possível definir uma política separada para a lâmpada que permite que ela publique em um espaço de tópico diferente. Por fim, é possível definir uma política para o aplicativo móvel que só permite que ele se conecte e se inscreva nos tópicos para o termostato e a lâmpada para receber mensagens desses dispositivos.

Aplique o princípio do privilégio mínimo e diminua o escopo das permissões por dispositivo o máximo possível. Todos os dispositivos ou usuários devem ter uma política de AWS IoT na AWS IoT que permita somente conectar-se a um ID de cliente conhecido, publicar e inscrever-se em um conjunto de tópicos identificado e fixo.

Use um segundoRegião da AWScomo backup

Considere armazenar uma cópia de seus dados em um segundoRegião da AWScomo um backup. Para obter mais informações, consulteRecuperação de desastresAWS IoT.

Usar provisionamento just-in-time

Criar e provisionar manualmente cada dispositivo pode ser demorado. O AWS IoT fornece uma forma de definir um modelo para provisionar dispositivos quando eles se conectam pela primeira vez ao AWS IoT. Para obter mais informações, consulteProvisionamento just-in-time (p. 815)

Permissões para executarAWS IoTTestes Device Advisor

O modelo de política a seguir mostra as permissões mínimas e a entidade do IAM necessárias para executarAWS IoTCasos de teste Device Advisor. Você precisará substituir*your-device-role-arn*Com a função de dispositivo Nome de recurso da Amazon (ARN) da função de dispositivo que você criou napré-requisitos.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": "your-device-role-arn",
            "Condition": {
                "StringEquals": {
                    "iam:PassedToService": "iotdeviceadvisor.amazonaws.com"
                }
            }
        },
        {
            "Sid": "VisualEditor1",
            "Effect": "Allow",
            "Action": [
                "execute-api:Invoke*",
                "iam:ListRoles",   // Required to list device roles in the Device Advisor
 console
                "iot:Connect",
                "iot:CreateJob",
                "iot:DeleteJob",
                "iot:DescribeCertificate",
                "iot:DescribeEndpoint",
                "iot:DescribeJobExecution",
                "iot:DescribeJob",
                "iot:DescribeThing",
                "iot:GetPendingJobExecutions",
                "iot:GetPolicy",
                "iot:ListAttachedPolicies",
                "iot:ListCertificates",
                "iot:ListPrincipalPolicies",
                "iot:ListThingPrincipals",
                "iot:ListThings",
                "iot:Publish",
                "iot:StartNextPendingJobExecution",
                "iot:UpdateJobExecution",
                "iot:UpdateThingShadow",
                "logs:CreateLogGroup",
                "logs:CreateLogStream",
                "logs:DescribeLogGroups",
                "logs:DescribeLogStreams",
                "logs:PutLogEvents",
                "logs:PutRetentionPolicy"
            ],
```

```
            "Resource": "*"
        },
        {
            "Sid": "VisualEditor2",
            "Effect": "Allow",
            "Action": "iotdeviceadvisor:*",
            "Resource": "*"
        }
    ]
}
```

Prevenção contra o problema entre serviços confusos para o Device Advisor

O problema de "confused deputy" é uma questão 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 o`aws:SourceArn`e`aws:SourceAccount`chaves de contexto de condição global em políticas de recursos para limitar as permissões que o Supervisor de dispositivos 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.

O valor de`aws:SourceArn`deve ser o ARN do recurso de definição de suíte. O recurso de definição de suíte refere-se ao conjunto de testes que você criou com o Device Advisor.

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

O exemplo a seguir mostra como você pode usar a`aws:SourceArn`e`aws:SourceAccount`chaves de contexto de condição global no Device Advisor para evitar o problema confuso do deputado.

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Sid": "ConfusedDeputyPreventionExamplePolicy",
    "Effect": "Allow",
    "Principal": {
      "Service": "iotdeviceadvisor.amazonaws.com"
    },
    "Action": "sts:AssumeRole",
    "Condition": {
      "ArnLike": {
        "aws:SourceArn": "arn:aws:iotdeviceadvisor:us-east-1:123456789012:suitedefinition/
ygp6rxa3tzvn"
      },
      "StringEquals": {
        "aws:SourceAccount": "123456789012"
      }
    }
```

```
    }
}
```

Treinamento e certificação da AWS

Faça o curso a seguir para aprender sobre os principais conceitos doAWS IoTSegurança:AWS IoT Primer de segurança.

Monitorar o AWS IoT

O monitoramento é uma parte importante da manutenção da confiabilidade, da disponibilidade e da performance do AWS IoT e de suas soluções da AWS.

É altamente recomendável que você colete dados de monitoramento de todas as partes da solução da AWS para facilitar a depuração de uma falha de vários pontos, caso ocorra. Comece criando um plano de monitoramento que responda às seguintes perguntas. Se não tiver certeza de como respondê-las, você ainda poderá continuar a habilitar o registro em log (p. 433) e estabelecer suas linhas de base de desempenho.

- Quais são seus objetivos de monitoramento?
- Quais recursos você vai monitorar?
- Com que frequência você vai monitorar esses recursos?
- Quais ferramentas de monitoramento você usará?
- Quem realizará o monitoramento das tarefas?
- Quem deve ser notificado quando algo der errado?

A próxima etapa é habilitar o registro em log (p. 433) e estabelecer uma linha de base de desempenho normal do AWS IoT no ambiente medindo o desempenho em vários momentos e em diferentes condições de carga. Conforme monitora o AWS IoT, mantenha os dados históricos de monitoramento para que você possa compará-los com os dados de desempenho atuais. Isso ajuda a identificar padrões normais e anomalias de desempenho e a criar métodos para tratar os problemas.

Para estabelecer seu desempenho de linha de base para a AWS IoT, monitore essas métricas para começar. Você sempre poderá monitorar mais métricas posteriormente.

- `PublishIn.Success` (p. 445)
- `PublishOut.Success` (p. 445)
- `Subscribe.Success` (p. 445)
- `Ping.Success` (p. 445)
- `Connect.Success` (p. 445)
- `GetThingShadow.Accepted` (p. 448)
- `UpdateThingShadow.Accepted` (p. 448)
- `DeleteThingShadow.Accepted` (p. 448)
- `RulesExecuted` (p. 444)

Os tópicos desta seção podem ajudar você a começar o registro e o monitoramento da AWS IoT.

Tópicos

- Configurar registro em log da AWS IoT (p. 433)
- MonitorAWS IoTalarmes e métricas usando o Amazon CloudWatch (p. 439)
- MonitorAWS IoTusando CloudWatch Logs (p. 454)
- Registre em log as chamadas à API do AWS IoT usando o AWS CloudTrail. (p. 475)

Configurar registro em log da AWS IoT

Você deve habilitar o registro em log usando o console do AWS IoT, a CLI ou a API antes de monitorar e registrar em log a atividade do AWS IoT.

É possível habilitar o registro em log para todo o AWS IoT ou somente grupos específicos. É possível configurar o registro em log do AWS IoT usando o console do AWS IoT, a CLI ou a API. No entanto, você deve usar a CLI ou a API para configurar o registro em log para grupos específicos de coisas.

Ao considerar como configurar o registro em log do AWS IoT, a configuração padrão determina como a atividade do AWS IoT será registrada, a menos que seja especificado de outra forma. No começo, talvez você queira obter logs detalhados com um nível de log (p. 438) padrão de `INFO` ou `DEBUG`. Depois que analisar os logs iniciais, você poderá alterar o nível de log padrão para um nível menos detalhado, como `WARN` ou `ERROR` e definir um nível de log mais detalhado específico do recursos em recursos que possam precisar de mais atenção. É possível alterar os níveis de log sempre que quiser.

Configurar a função e a política de registro em log

Antes de habilitar o login noAWS IoT, é necessário criar uma função do IAM e uma política que forneçaAWSpermissão para monitorarAWS IoTAtividade em seu nome.

> Note
>
> Antes de ativarAWS IoTRegistro em log, entenda o CloudWatch Registra permissões de acesso a Usuários com acesso a CloudWatch Os logs podem ver informações de depuração em seus dispositivos. Para obter mais informações, consulteAutenticação e controle de acesso para a Amazon CloudWatch Logs.
> Se você espera padrões de tráfego elevado emAWS IoT Coredevido ao teste de carga, considere desativar o registro de IoT para evitar o estrangulamento. Se alto tráfego for detectado, nosso serviço poderá desativar o registro em sua conta.

A seguir, mostra como criar uma função e uma política de log paraAWS IoT Corerecursos da AWS. Para obter informações sobre como você pode criar uma função e uma política de log do IAM paraAWS IoT Corepara LoRaWAN, consulteCriar uma política e uma função de registro em log paraAWS IoT Wireless (p. 1227).

Criar uma função de registro em log

Para criar uma função de log, abra aHub de funções do console do IAMe escolhaCriar função.

1. UNDERSelecionar o tipo de entidade confiável, escolhaAWSServiço,IoT.
2. UNDERSelecione seu caso de uso, escolhaIoTe, depois, escolhaPróximo: Permissions
3. Na página que exibe as políticas anexadas automaticamente à função de serviço, escolhaPróximo: Tagse, depois, escolhaPróximo: Análise.
4. Insira um Nome da função e uma Descrição da função e escolha Criar função.
5. Na lista deFunções do, encontre a função criada, abra-a e copie oARN de função(*`Logging-role-arn`*) para usar quando vocêConfigurar o registro em log padrão no AWS IoT (console) (p. 434).

Política de função de registro em log

Os documentos de políticas a seguir fornecem as políticas de função e de confiança que permitemAWS IoTPara enviar entradas de log ao CloudWatch Em seu nome. Se você também permitiuAWS IoT Corepara que o LoraWAN envie entradas de registro, você verá um documento de política criado para você que registra ambas as atividades. Para obter informações sobre como criar uma função e uma política de log do IAM paraAWS IoT Corepara LoRaWAN, consulteCriar uma política e uma função de registro em log paraAWS IoT Wireless (p. 1227).

> **Note**
>
> Esses documentos foram criados para você quando você criou a função de log. Os documentos têm variáveis,`${partition}`,`${region}`, e`${accountId}`, que você deve substituir por seus valores.

Política de função:

```
{
 "Version": "2012-10-17",
 "Statement": [
  {
   "Effect": "Allow",
   "Action": [
    "logs:CreateLogGroup",
    "logs:CreateLogStream",
    "logs:PutLogEvents",
    "logs:PutMetricFilter",
    "logs:PutRetentionPolicy"
   ],
   "Resource": [
    "arn:${partition}:logs:${region}:${accountId}:log-group:AWSIotLogsV2*"
   ]
  }
 ]
}
```

Política de confiança para registrar somenteAWS IoT Coreatividade:

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

Configurar o registro em log padrão no AWS IoT (console)

Esta seção descreve como usar o console do AWS IoT para configurar o registro em log para todo o AWS IoT. Para configurar o registro em log somente para grupos de coisas específicos, use a CLI ou a API. Para obter informações sobre como configurar o registro em log para grupos de coisas específicos, consulte Configurar registro em log de recursos específicos no AWS IoT (CLI) (p. 437).

Para usar o console do AWS IoT para configurar o registro em log padrão para todo o AWS IoT

1. Faça login no console do AWS IoT. Para obter mais informações, consulteAbrir oAWS IoTconsole (p. 21)
2. No painel de navegação à esquerda, escolha Settings (Configurações). Na seção Logs da página Settings (Configurações), escolha Edit (Editar).

A seção Logs exibe a função de registro em log e o nível de detalhamento usados por todo o AWS IoT.

3. NoConfigurar configuração de função, escolha oNível de verbosidadeque descreve oNível de detalhes (p. 438)das entradas de registro que você deseja que apareçam na CloudWatch logs.

4. Escolha Selecionar para especificar uma função que você criou em Criar uma função de registro em log (p. 433) ou Criar função para criar uma função a ser usada para o registro em log.
5. Escolha Update (Atualizar) para salvar suas alterações.

Depois que habilitar o registro em log, acesse Visualizar oAWS IoTlogs no CloudWatch console (p. 454) para saber mais sobre como visualizar as entradas de log.

Configurar registro em log padrão no AWS IoT (CLI)

Esta seção descreve como configurar o registro em log global para o AWS IoT usando a CLI.

> **Note**
>
> É necessário o nome de recurso da Amazon (ARN) da função que você deseja usar. Se você precisar criar uma função a ser usada para o registro em log, consulte Criar uma função de registro em log (p. 433) antes de continuar.
> O principal usado para chamar a API deve ter Transmitir permissões de função (p. 480) para sua função de registro em log.

Também é possível executar esse procedimento com a API usando os métodos na API da AWS que correspondam aos comandos da CLI mostrados aqui.

Para usar a CLI a fim de configurar o registro em log padrão para o AWS IoT

1. Use o comando set-v2-logging-options para definir as opções de registro em log para sua conta.

```
aws iot set-v2-logging-options \
    --role-arn logging-role-arn \
    --default-log-level log-level
```

   em que:

   --role-arn

   O ARN da função que concedeAWS IoTpermissão para gravar em seus logs no CloudWatch Logs.

   --default-log-level

   O nível de log (p. 438) a ser usado. Os valores válidos são `ERROR`, `WARN`, `INFO`, `DEBUG` ou `DISABLED`.

   --no-disable-all-logs

   Um parâmetro opcional que permite todos os registros em log do AWS IoT. Use esse parâmetro para habilitar o registro em log se ele estiver desabilitado no momento.

   --disable-all-logs

   Um parâmetro opcional que desativa todos os registros em do AWS IoT. Use esse parâmetro para desabilitar o registro em log se ele estiver habilitado no momento.

2. Use o comando get-v2-logging-options para obter as opções de registro em log atuais.

```
aws iot get-v2-logging-options
```

Depois que habilitar o registro em log, acesse Visualizar oAWS IoTlogs no CloudWatch console (p. 454) para saber mais sobre como visualizar as entradas de log.

> **Note**
>
> O AWS IoT continua a oferecer suporte a comandos mais antigos (set-logging-options e get-logging-options) para definir e obter o registro em log global na conta. Lembre-se de que, quando esses comandos são usados, os logs resultantes conterão texto sem formatação, em vez de cargas úteis JSON, e a latência dos logs geralmente será mais alta. Não haverá mais melhorias à implementação desses comandos mais antigos. Recomendamos que você use a versão "v2" para configurar suas opções de registro em log e, quando possível, altere quaisquer aplicativos legados que usam as versões mais antigas.

Configurar registro em log de recursos específicos no AWS IoT (CLI)

Esta seção descreve como configurar o registro em log específico de recursos para o AWS IoT usando a CLI. O registro em log de recursos específicos permite que você especifique um nível de log para um grupo de coisas (p. 272)específico.

Grupos de coisas podem conter outros grupos de coisas para criar um relacionamento hierárquico. Este procedimento descreve como configurar o registro em log de um único grupo de coisas. É possível aplicar esse procedimento ao grupo de coisas pai em uma hierarquia para configurar o registro em log para todos os grupos de coisas na hierarquia. Também é possível aplicar esse procedimento a um grupo de coisas filho para substituir a configuração de registro em log do grupo pai.

Além de grupos de coisas, você também pode registrar destinos como ID do cliente de um dispositivo, IP de origem e ID principal.

> **Note**
>
> É necessário o nome de recurso da Amazon (ARN) da função que você deseja usar. Se você precisar criar uma função a ser usada para o registro em log, consulte Criar uma função de registro em log (p. 433) antes de continuar.
> O principal usado para chamar a API deve ter Transmitir permissões de função (p. 480) para sua função de registro em log.

Também é possível executar esse procedimento com a API usando os métodos na API da AWS que correspondam aos comandos da CLI mostrados aqui.

Como usar a CLI para configurar o registro em log de recursos específicos para o AWS IoT

1. Use o comando set-v2-logging-options para definir as opções de registro em log para sua conta.

   ```
   aws iot set-v2-logging-options \
       --role-arn logging-role-arn \
       --default-log-level log-level
   ```

   em que:

   --role-arn

   O ARN da função que concedeAWS IoTpermissão para gravar em seus logs no CloudWatch Logs.

   --default-log-level

   O nível de log (p. 438) a ser usado. Os valores válidos são `ERROR`, `WARN`, `INFO`, `DEBUG` ou `DISABLED`.

   --no-disable-all-logs

   Um parâmetro opcional que permite todos os registros em log do AWS IoT. Use esse parâmetro para habilitar o registro em log se ele estiver desabilitado no momento.

   --disable-all-logs

   Um parâmetro opcional que desativa todos os registros em do AWS IoT. Use esse parâmetro para desabilitar o registro em log se ele estiver habilitado no momento.

2. Use o comando set-v2-logging-level para configurar o registro em log específico de recursos para um grupo de coisas.

   ```
   aws iot set-v2-logging-level \
   ```

```
              --log-target targetType=THING_GROUP,targetName=thing_group_name \
              --log-level log_level
```

--log-target

O tipo e o nome do recurso para o qual você está configurando o registro em log. O`target_type`O valor deve ser um dos seguintes:`THING_GROUP`|`CLIENT_ID`|`SOURCE_IP`|`PRINCIPAL_ID`. O valor do parâmetro log-target pode ser texto, como mostrado no exemplo de comando anterior, ou uma string JSON, como o exemplo a seguir.

```
aws iot set-v2-logging-level \
              --log-target '{"targetType": "THING_GROUP","targetName":
 "thing_group_name"}' \
              --log-level log_level
```

--log-level

O nível de log usado ao gerar logs para o recurso especificado. Os valores válidos são: DEBUG, INFO, ERROR, WARN e DISABLED

```
aws iot set-v2-logging-level \
              --log-target targetType=CLIENT_ID,targetName=ClientId1 \
              --log-level DEBUG
```

3. Use o comando list-v2-logging-levels para listar os níveis de registro em log configurados no momento.

```
aws iot list-v2-logging-levels
```

4. Usar adelete-v2-logging-levelPara excluir um nível de log específico de recursos, como os exemplos a seguir.

```
aws iot delete-v2-logging-level \
              --target-type "THING_GROUP" \
              --target-name "thing_group_name"
```

```
aws iot delete-v2-logging-level \
              --target-type=CLIENT_ID
              --target-name=ClientId1
```

--targetType

O`target_type`O valor deve ser um dos seguintes:`THING_GROUP`|`CLIENT_ID`|`SOURCE_IP`|`PRINCIPAL_ID`.

--targetName

O nome do grupo de coisas do qual remover o nível de registro em log.

Depois que habilitar o registro em log, acesse Visualizar oAWS IoTlogs no CloudWatch console (p. 454) para saber mais sobre como visualizar as entradas de log.

Níveis de log

Esses níveis de log determinam os eventos que são registrados em log e se aplicam aos níveis de log padrão e de recursos específicos.

ERROR

Qualquer erro que cause a falha de uma operação.

Os logs incluem apenas informações ERROR.

WARN

Tudo que possa causar inconsistências no sistema, mas que não cause falha na operação.

Os logs incluem informações ERROR e WARN.

INFO

Informações de alto nível sobre o fluxo das coisas.

Os logs incluem informações INFO, ERROR e WARN.

DEBUG

Informações que podem ser úteis ao depurar um problema.

Os logs incluem informações DEBUG, INFO, ERROR e WARN.

DISABLED

Todos os registros são desativados.

MonitorAWS IoTalarmes e métricas usando o Amazon CloudWatch

É possível monitorar o AWS IoT usando o CloudWatch, que coleta e processa dados brutos do AWS IoT e os transforma em métricas legíveis quase em tempo real. Essas estatísticas são registradas por um período de duas semanas, para que você possa acessar informações históricas e obter uma perspectiva melhor sobre como seu serviço ou aplicativo web estão se saindo. Por padrão,AWS IoTOs dados de métricas do são automaticamente enviados ao CloudWatch Em intervalos de um minuto. Para obter mais informações, consulteO que são o Amazon CloudWatch, Amazon CloudWatch Eventos e Amazon CloudWatch Logs?noAmazônia CloudWatch Guia do usuário do.

Usar métricas do AWS IoT

As métricas informadas pelo AWS IoT fornecem informações que você pode analisar de diferentes maneiras. Os seguintes casos de uso são baseados em um cenário com dez coisas que se conectam à Internet uma vez por dia. Cada dia:

- Dez coisas se conectam ao AWS IoT praticamente ao mesmo tempo.
- Cada coisa se inscreve um filtro do tópico e, em seguida, aguarda por uma hora antes de se desconectar. Durante esse período, as coisas se comunicam entre si e aprendem mais sobre o mundo.
- Cada coisa publica uma percepção com base nos dados recém-encontrados usando o `UpdateThingShadow`.
- Cada coisa se desconecta do AWS IoT.

Para ajudar você a começar, esses tópicos exploram algumas das perguntas que você pode ter.

- Como posso ser notificado se a conexão das minhas coisas não for bem-sucedida a cada dia? (p. 440)
- Como posso ser notificado se minhas coisas não estiverem publicando dados a cada dia? (p. 441)

- Como posso ser notificado se as atualizações da minha thing shadow estiverem sendo rejeitadas a cada dia? (p. 441)
- Como posso criar um CloudWatch alarme para Jobs? (p. 442)

Mais sobre CloudWatch Alarmes e métricas

- Criar CloudWatch Alarmes do para monitorarAWS IoT (p. 440)
- Métricas e dimensões do AWS IoT (p. 443)

Criar CloudWatch Alarmes do para monitorarAWS IoT

Você pode criar um CloudWatch O alarme do envia uma mensagem do Amazon SNS quando o alarme muda de estado. O alarme observa uma única métrica em um período especificado. Quando o valor da métrica exceder um determinado limite em vários períodos de tempo, uma ou mais ações serão executadas. A ação pode ser uma notificação enviada a um tópico do Amazon SNS ou uma política de Auto Scaling. Os alertas acionam ações apenas para alterações de estado mantidas. CloudWatch Os alarmes do não acionam ações simplesmente porque estão em um estado específico. O estado deve ter sido alterado e mantido por um número específico de períodos.

Os tópicos a seguir descrevem alguns exemplos de uso de alarmes do CloudWatch.

- Como posso ser notificado se a conexão das minhas coisas não for bem-sucedida a cada dia? (p. 440)
- Como posso ser notificado se minhas coisas não estiverem publicando dados a cada dia? (p. 441)
- Como posso ser notificado se as atualizações da minha thing shadow estiverem sendo rejeitadas a cada dia? (p. 441)
- Como posso criar um CloudWatch alarme para trabalhos? (p. 442)

Você pode ver todas as métricas do que CloudWatch alarmes podem monitorar emMétricas e dimensões do AWS IoT (p. 443).

Como posso ser notificado se a conexão das minhas coisas não for bem-sucedida a cada dia?

1. Criar um tópico do Amazon SNS chamado`things-not-connecting-successfully`E registre seu nome de recurso da Amazon (ARN). Esse procedimento se referirá ao ARN do seu tópico como *`sns-topic-arn`*.

   Para obter mais informações sobre como criar uma notificação do Amazon SNS, consulteConceitos básicos do Amazon SNS.
2. Crie o alerta.

```
aws cloudwatch put-metric-alarm \
    --alarm-name ConnectSuccessAlarm \
    --alarm-description "Alarm when my Things don't connect successfully" \
    --namespace AWS/IoT \
    --metric-name Connect.Success \
    --dimensions Name=Protocol,Value=MQTT \
    --statistic Sum \
    --threshold 10 \
    --comparison-operator LessThanThreshold \
    --period 86400 \
    --evaluation-periods 1 \
    --alarm-actions sns-topic-arn
```

3. Teste o alerta.

```
aws cloudwatch set-alarm-state --alarm-name ConnectSuccessAlarm --state-reason
 "initializing" --state-value OK
```

```
aws cloudwatch set-alarm-state --alarm-name ConnectSuccessAlarm --state-reason
 "initializing" --state-value ALARM
```

4. Verifique se o alarme aparece noConsole do CloudWatch.

Como posso ser notificado se minhas coisas não estiverem publicando dados a cada dia?

1. Criar um tópico do Amazon SNS chamado`things-not-publishing-data`E registre seu nome de recurso da Amazon (ARN). Esse procedimento se referirá ao ARN do seu tópico como *`sns-topic-arn`*.

   Para obter mais informações sobre como criar uma notificação do Amazon SNS, consulteConceitos básicos do Amazon SNS.
2. Crie o alerta.

```
aws cloudwatch put-metric-alarm \
    --alarm-name PublishInSuccessAlarm\
    --alarm-description "Alarm when my Things don't publish their data \
    --namespace AWS/IoT \
    --metric-name PublishIn.Success \
    --dimensions Name=Protocol,Value=MQTT \
    --statistic Sum \
    --threshold 10 \
    --comparison-operator LessThanThreshold \
    --period 86400 \
    --evaluation-periods 1 \
    --alarm-actions sns-topic-arn
```

3. Teste o alerta.

```
aws cloudwatch set-alarm-state --alarm-name PublishInSuccessAlarm --state-reason
 "initializing" --state-value OK
```

```
aws cloudwatch set-alarm-state --alarm-name PublishInSuccessAlarm --state-reason
 "initializing" --state-value ALARM
```

4. Verifique se o alarme aparece noConsole do CloudWatch.

Como posso ser notificado se as atualizações da minha thing shadow estiverem sendo rejeitadas a cada dia?

1. Criar um tópico do Amazon SNS chamado`things-shadow-updates-rejected`E registre seu nome de recurso da Amazon (ARN). Esse procedimento se referirá ao ARN do seu tópico como *`sns-topic-arn`*.

   Para obter mais informações sobre como criar uma notificação do Amazon SNS, consulteConceitos básicos do Amazon SNS.
2. Crie o alerta.

```
aws cloudwatch put-metric-alarm \
    --alarm-name UpdateThingShadowSuccessAlarm \
    --alarm-description "Alarm when my Things Shadow updates are getting rejected" \
    --namespace AWS/IoT \
    --metric-name UpdateThingShadow.Success \
    --dimensions Name=Protocol,Value=MQTT \
    --statistic Sum \
    --threshold 10 \
    --comparison-operator LessThanThreshold \
    --period 86400 \
    --unit Count \
    --evaluation-periods 1 \
    --alarm-actions sns-topic-arn
```

3. Teste o alerta.

```
aws cloudwatch set-alarm-state --alarm-name UpdateThingShadowSuccessAlarm --state-
reason "initializing" --state-value OK
```

```
aws cloudwatch set-alarm-state --alarm-name UpdateThingShadowSuccessAlarm --state-
reason "initializing" --state-value ALARM
```

4. Verifique se o alarme aparece noConsole do CloudWatch.

Como posso criar um CloudWatch alarme para trabalhos?

O serviço Jobs fornece CloudWatch Métricas para você monitorar seus trabalhos. É possível criar alarmes do CloudWatch para monitorar quaisquer Métricas de trabalhos (p. 449).

O comando a seguir cria um CloudWatch Alarme para monitorar o número total de execuções de Job com falha para o trabalho*SampleOTAJob*E o notifica quando há falha em mais de 20 execuções de trabalho. O alarme monitora a métrica `FailedJobExecutionTotalCount` do Jobs verificando o valor relatado a cada 300 segundos. Ele é ativado quando um único valor relatado é maior que 20, o que significa que houve mais de 20 execuções de trabalho com falha desde que o trabalho foi iniciado. Quando o alarme dispara, ele envia uma notificação para o tópico do Amazon SNS fornecido.

```
aws cloudwatch put-metric-alarm \
    --alarm-name TotalFailedJobExecution-SampleOTAJob \
    --alarm-description "Alarm when total number of failed job execution exceeds the
 threshold for SampleOTAJob" \
    --namespace AWS/IoT \
    --metric-name FailedJobExecutionTotalCount \
    --dimensions Name=JobId,Value=SampleOTAJob \
    --statistic Sum \
    --threshold 20 \
    --comparison-operator GreaterThanThreshold \
    --period 300 \
    --unit Count \
    --evaluation-periods 1 \
    --alarm-actions arn:aws:sns:<AWS_REGION>:<AWS_ACCOUNT_ID>:SampleOTAJob-has-too-many-
failed-job-ececutions
```

O comando a seguir cria um CloudWatch Alarme para monitorar o número de execuções de Job com falha para o trabalho*SampleOTAJob*em um determinado período. Ele, então, o notifica quando há falha em mais de cinco execuções do trabalho durante esse período. O alarme monitora a métrica `FailedJobExecutionCount` do Jobs verificando o valor relatado a cada 3.600 segundos. Ele é ativado quando um único valor relatado é maior que 5, o que significa que houve mais de 5 execuções de trabalho

com falha na última hora. Quando o alarme dispara, ele envia uma notificação para o tópico do Amazon SNS fornecido.

```
aws cloudwatch put-metric-alarm \
    --alarm-name FailedJobExecution-SampleOTAJob \
    --alarm-description "Alarm when number of failed job execution per hour exceeds the
 threshold for SampleOTAJob" \
    --namespace AWS/IoT \
    --metric-name FailedJobExecutionCount \
    --dimensions Name=JobId,Value=SampleOTAJob \
    --statistic Sum \
    --threshold 5 \
    --comparison-operator GreaterThanThreshold \
    --period 3600 \
    --unit Count \
    --evaluation-periods 1 \
    --alarm-actions arn:aws:sns:<AWS_REGION>:<AWS_ACCOUNT_ID>:SampleOTAJob-has-too-many-
failed-job-ececutions-per-hour
```

Métricas e dimensões do AWS IoT

Ao interagir com a AWS IoT, o serviço envia as seguintes métricas e dimensões para o CloudWatch a cada minuto. É possível usar os procedimentos a seguir para visualizar as métricas do AWS IoT.

Para exibir métricas (console do CloudWatch)

As métricas são agrupadas primeiro pelo namespace do serviço e, em seguida, por várias combinações de dimensão dentro de cada namespace.

1. Abra o console do CloudWatch.
2. No painel de navegação, selecioneMétricase, depois, escolhaTodas as métricas.
3. NoNavegar, procureAWS IoTpara exibir a lista de métricas.

Para visualizar métricas (CLI)

- Em um prompt de comando, use o seguinte comando:

```
aws cloudwatch list-metrics --namespace "AWS/IoT"
```

O CloudWatch exibe os seguintes grupos de métricas paraAWS IoT:

- AWS IoT métricas (p. 444)
- AWS IoT CoreMétricas do provedor de credenciais (p. 444)
- Métricas de regra (p. 444)
- Métricas de ação da regra (p. 445)
- Métricas específicas da ação HTTP (p. 445)
- Métricas do agente de mensagens (p. 445)
- Métricas do device shadow (p. 448)
- Métricas de trabalhos (p. 449)
- Métricas de auditoria do Device Defender (p. 450)
- Métricas do Device Defender Detect (p. 451)

- Métricas de provisionamento de dispositivos (p. 451)
- Métricas de indexação de frota (p. 453)
- Dimensões para métricas (p. 453)

AWS IoT métricas

Métrica	Descrição
`AddThingToDynamicThingGroupsFailed`	O número de eventos de falha associados à adição de uma coisa a um grupo dinâmico. A dimensão `DynamicThingGroupName` contém o nome dos grupos dinâmicos que falharam ao adicionar coisas.
`NumLogBatchesFailedToPublishThrottled`	O lote singular de eventos de log dentro do lote que falhou na publicação devido a erros de limitação.
`NumLogEventsFailedToPublishThrottled`	O número de eventos de log dentro do lote que falharam na publicação devido a erros de limitação.

AWS IoT CoreMétricas do provedor de credenciais

Métrica	Descrição
`CredentialExchangeSuccess`	O número de bem-sucedidos`AssumeRoleWithCertificate`Solicitações paraAWS IoT Coreprovedor de credenciais do.

Métricas de regra

Métrica	Descrição
`ParseError`	O número de erros de análise JSON encontrados em mensagens publicadas em um tópico no qual uma regra está escutando. A dimensão `RuleName` contém o nome da regra.
`RuleMessageThrottled`	O número de mensagens limitadas pelo mecanismo de regras devido a comportamentos mal-intencionados ou porque o número de mensagens excede o limite de controle de fluxo do mecanismo de regras. A dimensão `RuleName` contém o nome da regra a ser acionada.
`RuleNotFound`	Não foi possível encontrar a regra a ser acionada. A dimensão `RuleName` contém o nome da regra.
`RulesExecuted`	O número de regras do AWS IoT executadas.
`TopicMatch`	O número de mensagens de entrada publicadas em um tópico no qual uma regra está escutando. A dimensão `RuleName` contém o nome da regra.

Métricas de ação da regra

Métrica	Descrição
`Failure`	O número de invocações da ação de regra com falha. A dimensão `RuleName` contém o nome da regra que especifica a ação. A dimensão `ActionType` contém o tipo de ação que foi invocada.
`Success`	O número de invocações da ação de regra bem-sucedidas. A dimensão `RuleName` contém o nome da regra que especifica a ação. A dimensão `ActionType` contém o tipo de ação que foi invocada.
`ErrorActionFailure`	O número de ações de erro com falha. A dimensão `RuleName` contém o nome da regra que especifica a ação. A dimensão `ActionType` contém o tipo de ação que foi invocada.
`ErrorActionSuccess`	O número de ações de erro bem-sucedidas. A dimensão `RuleName` contém o nome da regra que especifica a ação. A dimensão `ActionType` contém o tipo de ação que foi invocada.

Métricas específicas da ação HTTP

Métrica	Descrição
`HttpCode_Other`	Gerado se o código de status da resposta do serviço/aplicativo web downstream não for 2xx, 4xx ou 5xx.
`HttpCode_4XX`	Gerado se o código de status da resposta do serviço/aplicativo web downstream estiver entre 400 e 499.
`HttpCode_5XX`	Gerado se o código de status da resposta do serviço/aplicativo web downstream estiver entre 500 e 599.
`HttpInvalidUrl`	Gerado se um URL de endpoint, após a substituição dos modelos de substituição, não começar com `https://`.
`HttpRequestTimeout`	Gerado se o serviço/aplicativo web downstream não retornar resposta dentro do limite de tempo limite de solicitação. Para obter mais informações, consulte Cotas de serviço.
`HttpUnknownHost`	Gerado se o URL for válido, mas o serviço não existe ou é inacessível.

Métricas do agente de mensagens

> **Note**
>
> As métricas do agente de mensagens são exibidas no CloudWatch console sobMétricas de Protocol.

Métrica	Descrição
`Connect.AuthError`	O número de solicitações de conexão que não puderam ser autorizadas pelo agente de mensagens. A dimensão `Protocol` contém o protocolo usado para enviar a mensagem `CONNECT`.
`Connect.ClientError`	O número de solicitações de conexão rejeitadas porque a mensagem MQTT não atendeu aos requisitos definidos em AWS IoTCotas do (p. 1298). A dimensão `Protocol` contém o protocolo usado para enviar a mensagem `CONNECT`.
`Connect.ClientIDThrottle`	O número de solicitações de conexão limitadas porque o cliente excedeu a taxa de solicitação da conexão permitida para um ID de cliente específico. A dimensão `Protocol` contém o protocolo usado para enviar a mensagem `CONNECT`.
`Connect.ServerError`	O número de solicitações de conexão com falha porque ocorreu um erro interno. A dimensão `Protocol` contém o protocolo usado para enviar a mensagem `CONNECT`.
`Connect.Success`	O número de conexões bem-sucedidas com o agente de mensagens. A dimensão `Protocol` contém o protocolo usado para enviar a mensagem `CONNECT`.
`Connect.Throttle`	O número de solicitações de conexão que foram limitadas porque a conta excedeu a taxa de solicitações de conexão permitida. A dimensão `Protocol` contém o protocolo usado para enviar a mensagem `CONNECT`.
`Ping.Success`	O número de mensagens ping recebidas pelo agente de mensagens. A dimensão `Protocol` contém o protocolo usado para enviar a mensagem ping.
`PublishIn.AuthError`	O número de solicitações de publicação que o agente de mensagens não conseguiu autorizar. A dimensão `Protocol` contém o protocolo usado para publicar a mensagem.
`PublishIn.ClientError`	O número de solicitações de publicação rejeitadas pelo agente de mensagens porque a mensagem não atendeu aos requisitos definidos em AWS IoTCotas do (p. 1298). A dimensão `Protocol` contém o protocolo usado para publicar a mensagem.
`PublishIn.ServerError`	O número de solicitações de publicação que o agente de mensagens deixou de processar por causa de um erro interno ocorrido. A dimensão `Protocol` contém o protocolo usado para enviar a mensagem `PUBLISH`.
`PublishIn.Success`	O número de solicitações de publicação processadas com êxito pelo agente de mensagens. A dimensão `Protocol` contém o protocolo usado para enviar a mensagem `PUBLISH`.
`PublishIn.Throttle`	O número de solicitações de publicação que foram limitadas porque o cliente excedeu a taxa de mensagens

Métrica	Descrição
	recebidas permitida. A dimensão `Protocol` contém o protocolo usado para enviar a mensagem `PUBLISH`.
`PublishOut.AuthError`	O número de solicitações de publicação feitas pelo agente de mensagens que não puderam ser autorizadas pelo AWS IoT. A dimensão `Protocol` contém o protocolo usado para enviar a mensagem `PUBLISH`.
`PublishOut.ClientError`	O número de solicitações de publicação feitas pelo agente de mensagens que foram rejeitadas porque a mensagem não atendeu aos requisitos definidos em AWS IoTCotas do (p. 1298). A dimensão `Protocol` contém o protocolo usado para enviar a mensagem `PUBLISH`.
`PublishOut.Success`	O número de solicitações de publicação feitas com êxito pelo agente de mensagens. A dimensão `Protocol` contém o protocolo usado para enviar a mensagem `PUBLISH`.
`PublishOut.Throttle`	O número de solicitações de publicação que foram limitadas porque o cliente excedeu a taxa de mensagens recebidas permitida. A dimensão `Protocol` contém o protocolo usado para enviar a mensagem `PUBLISH`.
`PublishRetained.AuthError`	O número de solicitações de publicação com `oRETAIN`Conjunto de sinalizadores que o agente de mensagens não conseguiu autorizar. A dimensão `Protocol` contém o protocolo usado para enviar a mensagem `PUBLISH`.
`PublishRetained.ServerError`	O número de solicitações de publicação retidas que o agente de mensagens deixou de processar por causa de um erro interno ocorrido. A dimensão `Protocol` contém o protocolo usado para enviar a mensagem `PUBLISH`.
`PublishRetained.Success`	O número de solicitações de publicação com `oRETAIN`Conjunto de sinalizadores que foram processados com êxito pelo agente de mensagens. A dimensão `Protocol` contém o protocolo usado para enviar a mensagem `PUBLISH`.
`PublishRetained.Throttle`	O número de solicitações de publicação com `oRETAIN`Conjunto de sinalizadores que foram limitados porque o cliente excedeu a taxa de mensagens recebidas permitida. A dimensão `Protocol` contém o protocolo usado para enviar a mensagem `PUBLISH`.
`Subscribe.AuthError`	O número de solicitações de assinatura feitas por um cliente que não puderam ser autorizadas. A dimensão `Protocol` contém o protocolo usado para enviar a mensagem `SUBSCRIBE`.

Métrica	Descrição
`Subscribe.ClientError`	O número de solicitações de assinatura que foram rejeitadas porque a mensagem `SUBSCRIBE` não atendeu aos requisitos definidos em AWS IoTCotas do (p. 1298). A dimensão `Protocol` contém o protocolo usado para enviar a mensagem `SUBSCRIBE`.
`Subscribe.ServerError`	O número de solicitações de assinatura que foram rejeitadas porque ocorreu um erro interno. A dimensão `Protocol` contém o protocolo usado para enviar a mensagem `SUBSCRIBE`.
`Subscribe.Success`	O número de solicitações de assinatura que foram processadas com êxito pelo agente de mensagens. A dimensão `Protocol` contém o protocolo usado para enviar a mensagem `SUBSCRIBE`.
`Subscribe.Throttle`	O número de solicitações de assinatura que foram limitadas porque o cliente excedeu a taxa de solicitação da assinatura permitida. A dimensão `Protocol` contém o protocolo usado para enviar a mensagem `SUBSCRIBE`.
`Unsubscribe.ClientError`	O número de solicitações de cancelamento da assinatura que foram rejeitadas porque a mensagem `UNSUBSCRIBE` não atendeu aos requisitos definidos em AWS IoTCotas do (p. 1298). A dimensão `Protocol` contém o protocolo usado para enviar a mensagem `UNSUBSCRIBE`.
`Unsubscribe.ServerError`	O número de solicitações de cancelamento da assinatura que foram rejeitadas porque ocorreu um erro interno. A dimensão `Protocol` contém o protocolo usado para enviar a mensagem `UNSUBSCRIBE`.
`Unsubscribe.Success`	O número de solicitações de cancelamento da assinatura que foram processadas com êxito pelo agente de mensagens. A dimensão `Protocol` contém o protocolo usado para enviar a mensagem `UNSUBSCRIBE`.
`Unsubscribe.Throttle`	O número de solicitações de cancelamento da assinatura que foram rejeitadas porque o cliente excedeu a taxa de solicitação de cancelamento da assinatura permitida. A dimensão `Protocol` contém o protocolo usado para enviar a mensagem `UNSUBSCRIBE`.

Métricas do device shadow

> **Note**
>
> As métricas de sombra do dispositivo são exibidas no CloudWatch console sobMétricas de Protocol.

Métrica	Descrição
`DeleteThingShadow.Accepted`	O número de solicitações `DeleteThingShadow` processadas com êxito. A dimensão `Protocol` contém o protocolo usado para criar a solicitação.
`GetThingShadow.Accepted`	O número de solicitações `GetThingShadow` processadas com êxito. A dimensão `Protocol` contém o protocolo usado para criar a solicitação.
`ListThingShadow.Accepted`	O número de solicitações `ListThingShadow` processadas com êxito. A dimensão `Protocol` contém o protocolo usado para criar a solicitação.
`UpdateThingShadow.Accepted`	O número de solicitações `UpdateThingShadow` processadas com êxito. A dimensão `Protocol` contém o protocolo usado para criar a solicitação.

Métricas de trabalhos

Métrica	Descrição
`CanceledJobExecutionCount`	O número de execuções de trabalho cujo o status mudou `CANCELED` em um período que foi determinado pelo CloudWatch. (Para obter mais informações sobre CloudWatch métricas, consulteMétricas do Amazon CloudWatch.) A dimensão `JobId` contém o ID do trabalho.
`CanceledJobExecutionTotalCount`	O número total de execuções do trabalho cujo status for `CANCELED` para o trabalho determinado. A dimensão `JobId` contém o ID do trabalho.
`ClientErrorCount`	O número de erros do cliente gerados ao executar o trabalho. A dimensão `JobId` contém o ID do trabalho.
`FailedJobExecutionCount`	O número de execuções de trabalho cujo o status mudou `FAILED` em um período que foi determinado pelo CloudWatch. (Para obter mais informações sobre CloudWatch métricas, consulteAmazônia CloudWatch Métricas.) A dimensão `JobId` contém o ID do trabalho.
`FailedJobExecutionTotalCount`	O número total de execuções do trabalho cujo status for `FAILED` para o trabalho determinado. A dimensão `JobId` contém o ID do trabalho.
`InProgressJobExecutionCount`	O número de execuções de trabalho cujo o status mudou `IN_PROGRESS` em um período que foi determinado pelo CloudWatch. (Para obter mais informações sobre CloudWatch métricas, consulteMétricas do Amazon CloudWatch.) A dimensão `JobId` contém o ID do trabalho.
`InProgressJobExecutionTotalCount`	O número total de execuções do trabalho cujo status for `IN_PROGRESS` para o trabalho determinado. A dimensão `JobId` contém o ID do trabalho.

Métrica	Descrição
`RejectedJobExecutionTotalCount`	O número total de execuções do trabalho cujo status for `REJECTED` para o trabalho determinado. A dimensão `JobId` contém o ID do trabalho.
`RemovedJobExecutionTotalCount`	O número total de execuções do trabalho cujo status for `REMOVED` para o trabalho determinado. A dimensão `JobId` contém o ID do trabalho.
`QueuedJobExecutionCount`	O número de execuções de trabalho cujo o status mudou `QUEUED` em um período que foi determinado pelo CloudWatch. (Para obter mais informações sobre CloudWatch métricas, consulteAmazônia CloudWatch Métricas.) A dimensão `JobId` contém o ID do trabalho.
`QueuedJobExecutionTotalCount`	O número total de execuções do trabalho cujo status for `QUEUED` para o trabalho determinado. A dimensão `JobId` contém o ID do trabalho.
`RejectedJobExecutionCount`	O número de execuções de trabalho cujo o status mudou `REJECTED` em um período que foi determinado pelo CloudWatch. (Para obter mais informações sobre CloudWatch métricas, consulteMétricas do Amazon CloudWatch.) A dimensão `JobId` contém o ID do trabalho.
`RemovedJobExecutionCount`	O número de execuções de trabalho cujo o status mudou `REMOVED` em um período que foi determinado pelo CloudWatch. (Para obter mais informações sobre CloudWatch métricas, consulteMétricas do Amazon CloudWatch.) A dimensão `JobId` contém o ID do trabalho.
`ServerErrorCount`	O número de erros do servidor gerados ao executar o trabalho. A dimensão `JobId` contém o ID do trabalho.
`SuccededJobExecutionCount`	O número de execuções de trabalho cujo o status mudou `SUCCESS` em um período que foi determinado pelo CloudWatch. (Para obter mais informações sobre CloudWatch métricas, consulteMétricas do Amazon CloudWatch.) A dimensão `JobId` contém o ID do trabalho.
`SuccededJobExecutionTotalCount`	O número total de execuções do trabalho cujo status for `SUCCESS` para o trabalho determinado. A dimensão `JobId` contém o ID do trabalho.

Métricas de auditoria do Device Defender

Métrica	Descrição
`NonCompliantResources`	O número de recursos que foram identificados como não compatíveis com uma verificação. O sistema informará o número de recursos que estavam fora da conformidade para cada verificação de cada auditoria feita.

Métrica	Descrição
`ResourcesEvaluated`	O número de recursos que foram avaliados para conformidade. O sistema informará o número de recursos que foram avaliados para cada verificação de cada auditoria feita.

Métricas do Device Defender Detect

Métrica	Descrição
`Violations`	O número de violações novas dos comportamentos de perfil de segurança que foram localizadas desde a última vez que uma avaliação foi feita. O sistema informará o número de violações novas da conta, para um perfil de segurança específico e para um comportamento específico de um perfil de segurança específico.
`ViolationsCleared`	O número de violações dos comportamentos de perfil de segurança que foram resolvidos desde a última vez que uma avaliação foi feita. O sistema informará o número de violações resolvidas da conta, para um perfil de segurança específico e para um comportamento específico de um perfil de segurança específico.
`ViolationsInvalidated`	O número de violações de comportamentos de perfil de segurança cujas informações não estão mais disponíveis desde a última vez que uma avaliação foi feita (porque o dispositivo de relatório parou de gerar relatórios ou não está mais sendo monitorado por algum motivo). O sistema informará o número de violações invalidadas da conta inteira, para um perfil de segurança específico e para um comportamento específico de um perfil de segurança específico.

Métricas de provisionamento de dispositivos

AWS IoTMétricas de provisionamento de frota

Métrica	Descrição
`ApproximateNumberOfThingsRegistered`	A contagem de itens que foram registrados pelo Provisionamento de frotas. Embora a contagem seja geralmente precisa, a arquitetura distribuída doAWS IoT Coredificulta a manutenção de uma contagem precisa de coisas registradas. A estatística a ser usada para essa métrica é: • máximopara relatar o número total de coisas que foram registradas. Para uma contagem de coisas registradas durante a janela

Métrica	Descrição
	de agregação do CloudWatch, consulte `oRegisterThingFailed`Métrica do. Dimensões: ClaimCertificateID (p. 453)
`CreateKeysAndCertificateFailed`	O número de falhas que ocorreram por chamadas para `oCreateKeysAndCertificate`MQTT API. A métrica é emitida nos casos Sucesso (valor = 0) e Falha (valor = 1). Essa métrica pode ser usada para rastrear o número de certificados criados e registrados durante as janelas de agregação suportadas pelo CloudWatch, como 5 min. ou 1 hora. As estatísticas disponíveis para essa métrica são: • Somapara relatar o número de chamadas com falha. • SampleCountpara relatar o número total de chamadas bem-sucedidas e com falha.
`CreateCertificateFromCsrFailed`	O número de falhas que ocorreram por chamadas para `oCreateCertificateFromCsr`MQTT API. A métrica é emitida nos casos Sucesso (valor = 0) e Falha (valor = 1). Essa métrica pode ser usada para rastrear o número de coisas registradas durante as janelas de agregação suportadas pelo CloudWatch, como 5 min. ou 1 hora. As estatísticas disponíveis para essa métrica são: • Somapara relatar o número de chamadas com falha. • SampleCountpara relatar o número total de chamadas bem-sucedidas e com falha.
`RegisterThingFailed`	O número de falhas que ocorreram por chamadas para `oRegisterThing`MQTT API. A métrica é emitida nos casos Sucesso (valor = 0) e Falha (valor = 1). Essa métrica pode ser usada para rastrear o número de coisas registradas durante as janelas de agregação suportadas pelo CloudWatch, como 5 min. ou 1 hora. Para o número total de coisas registradas, consulte `oApproximateNumberOfThingsRegistered`Métrica do. As estatísticas disponíveis para essa métrica são: • Somapara relatar o número de chamadas com falha. • SampleCountpara relatar o número total de chamadas bem-sucedidas e com falha. Dimensões: TemplateName (p. 453)

Métricas de provisionamento just-in-time

Métrica	Descrição
`ProvisionThing.ClientError`	O número de vezes em que houve falha no provisionamento de um dispositivo devido a um erro do cliente. Por exemplo, a política especificada no modelo não existia.
`ProvisionThing.ServerError`	O número de vezes em que houve falha no provisionamento de um dispositivo devido a um erro do servidor. Os clientes podem tentar provisionar novamente o dispositivo após a espera e podem entrar em contatoAWS IoTSe o problema permanecer o mesmo.
`ProvisionThing.Success`	O número de vezes em que um dispositivo foi provisionado com êxito.

Métricas de indexação de frota

AWS IoTMétricas de indexação de frota

Métrica	Descrição
`DeviceDefenderThingViolationsEventSizeLimitExceeded`	O tamanho total dos dados de uma coisa processada pela indexação da frota é limitado a 32 KB. Quando esse limite for violado por uma coisa devido a um evento de violações do Device Defender, o `DeviceDefenderThingViolationsEventSizeLimitExceeded` tipo de evento será emitido.
`NamedShadowEventSizeLimitExceeded`	O tamanho total dos dados de uma coisa processada pela indexação da frota é limitado a 32 KB. Quando esse limite é violado por uma coisa devido a um evento sombra nomeado, o `NamedShadowEventSizeLimitExceeded` tipo de evento será emitido.
`NamedShadowCountForDynamicGroupQueryLimitExceeded`	No máximo 5 sombras nomeadas por coisa são processadas para termos de consulta que não são específicos da fonte de dados em grupos de coisas dinâmicas. Quando esse limite é violado por uma coisa, o `NamedShadowCountForDynamicGroupQueryLimitExceeded` tipo de evento será emitido.

Dimensões para métricas

As métricas usam o namespace e fornecem métricas para as dimensões a seguir.

Dimensão	Descrição
`ActionType`	O tipo de ação (p. 485) especificado pela regra que acionou a solicitação.

Dimensão	Descrição
`BehaviorName`	O nome do comportamento do perfil de segurança do Device Defender Detect que está sendo monitorado.
`ClaimCertificateId`	O`certificateId`da reivindicação usada para provisionar os dispositivos.
`CheckName`	O nome da verificação de auditoria do Device Defender cujos resultados estão sendo monitorados.
`JobId`	O ID do trabalho cujo progresso ou mensagem de erro/êxito de conexão estiverem sendo monitorados.
`Protocol`	O protocolo usado para fazer a solicitação. Os valores válidos são: MQTT ou HTTP
`RuleName`	O nome da regra disparada pela solicitação.
`ScheduledAuditName`	O nome da auditoria programada do Device Defender cujos resultados das verificações estão sendo monitorados. Terá o valor `OnDemand` se os resultados informados forem para uma auditoria que foi feita sob demanda.
`SecurityProfileName`	O nome do perfil de segurança do Device Defender Detect cujos comportamentos estão sendo monitorados.
`TemplateName`	O nome do modelo de provisionamento.

MonitorAWS IoTusando CloudWatch Logs

Quando o registro em log da AWS IoT (p. 433) está habilitado, a AWS IoT envia eventos de progresso sobre cada mensagem conforme ela passa dos dispositivos para o agente de mensagens e o mecanismo de regras. NoConsole do CloudWatch, CloudWatch Os logs do aparecem em um grupo de logs chamadoAWSIotLogs.

Para obter mais informações sobre CloudWatch Registros, consulteCloudWatch Logs. Para obter informações sobre compatívelAWS IoT CloudWatch Registros, consulteCloudWatchAWS IoTEntradas de log do do (p. 455).

Visualizar oAWS IoTlogs no CloudWatch console

> **Note**
>
> O`AWSIotLogsV2`grupo de logs não está visível na CloudWatch console até:
>
> - Você ativou o loginAWS IoT. Para obter mais informações sobre como habilitar o loginAWS IoT, consulteConfigurar registro em log da AWS IoT (p. 433)
> - Algumas entradas de registro foram escritas porAWS IoToperações.

Para visualizar osAWS IoTlogs no CloudWatch console

1. Navegue parahttps://console.aws.amazon.com/cloudwatch/. No painel de navegação, escolha Log groups(Grupos de logs).

2. Na caixa de texto Filter (Filtro), digite **`AWSIotLogsV2`** e, depois, pressione Enter.
3. Clique duas vezes no grupo de logs `AWSIotLogsV2`.
4. SelecionePesquisar tudo. Uma lista completa dos logs da AWS IoT gerados para sua conta é exibida.
5. Escolha o ícone de expansão para ver um fluxo individual.

Você também pode inserir uma consulta na caixa de texto Filter events (Filtrar eventos). Estas são algumas consultas interessantes para experimentar:

- `{ $.logLevel = "INFO" }`

  Encontre todos os logs que têm um nível de log de `INFO`.
- `{ $.status = "Success" }`

  Encontre todos os logs que têm um status de `Success`.
- `{ $.status = "Success" && $.eventType = "GetThingShadow" }`

  Encontre todos os logs que têm um status de `Success` e um tipo de evento de `GetThingShadow`.

Para obter mais informações sobre a criação de expressões de filtro, consulteConsultas do CloudWatch Logs.

CloudWatchAWS IoTEntradas de log do do

Cada componente do AWS IoT gera suas próprias entradas de log. Cada entrada de log tem um `eventType` que especifica a operação que fez com que a entrada de log fosse gerada. Esta seção descreve as entradas de log geradas pelo seguinteAWS IoTComponentes do. Para obter informações sobre oAWS IoT Corepara monitoramento LoRaWAN, consulteExibir CloudWatch AWS IoT WirelessEntradas de log do do (p. 1238).

Tópicos

- Entradas de log do agente de mensagens (p. 455)
- Entradas de log do Device Shadow (p. 461)
- Entradas de log do mecanismo de regras (p. 462)
- Entradas de log de trabalhos (p. 467)
- Entradas de log de provisionamento de dispositivos (p. 471)
- Entradas de log de grupo de coisas dinâmico (p. 472)
- Entradas de log de indexação de fro (p. 473)
- Comum CloudWatch Atributos logs (p. 474)

Entradas de log do agente de mensagens

O agente de mensagens do AWS IoT gera logs para os seguintes eventos:

Tópicos

- Entrada de log de conexão (p. 456)
- Entrada de log de desconexão (p. 456)
- Entrada de log GetRetainedMessage (p. 457)
- Entrada de log ListRetainedMessage (p. 458)
- Entrada de log de publicação de entrada (p. 458)

- Entrada de log de publicação de saída (p. 459)
- Entrada de log de assinatura (p. 460)

Entrada de log de conexão

O agente de mensagens do AWS IoT gera uma entrada de log com um `eventType` de `Connect` quando um cliente MQTT se conecta.

Exemplo de entrada de log de conexão

```
{
    "timestamp": "2017-08-10 15:37:23.476",
    "logLevel": "INFO",
    "traceId": "20b23f3f-d7f1-feae-169f-82263394fbdb",
    "accountId": "123456789012",
    "status": "Success",
    "eventType": "Connect",
    "protocol": "MQTT",
    "clientId": "abf27092886e49a8a5c1922749736453",
    "principalId": "145179c40e2219e18a909d896a5340b74cf97a39641beec2fc3eeafc5a932167",
    "sourceIp": "205.251.233.181",
    "sourcePort": 13490
}
```

Além do Comum CloudWatch Atributos logs (p. 474), as entradas de log `Connect` contêm os seguintes atributos:

clientId

O ID do cliente que está fazendo a solicitação.

principalId

O ID da entidade principal que está fazendo a solicitação.

protocolo

O protocolo usado ao criar a solicitação. Os valores válidos são `MQTT` ou `HTTP`.

sourceIp

O endereço IP no qual a solicitação foi originada.

sourcePort

A porta em que a solicitação foi originada.

Entrada de log de desconexão

O agente de mensagens do AWS IoT gera uma entrada de log com um `eventType` de `Disconnect` quando um cliente MQTT se desconecta.

Exemplo de entrada de log de desconexão

```
{
    "timestamp": "2017-08-10 15:37:23.476",
    "logLevel": "INFO",
    "traceId": "20b23f3f-d7f1-feae-169f-82263394fbdb",
    "accountId": "123456789012",
    "status": "Success",
```

```
    "eventType": "Disconnect",
    "protocol": "MQTT",
    "clientId": "abf27092886e49a8a5c1922749736453",
    "principalId": "145179c40e2219e18a909d896a5340b74cf97a39641beec2fc3eeafc5a932167",
    "sourceIp": "205.251.233.181",
    "sourcePort": 13490,
    "disconnectReason": "CLIENT_INITIATED_DISCONNECT"
}
```

Além do Comum CloudWatch Atributos logs (p. 474), as entradas de log `Disconnect` contêm os seguintes atributos:

clientId

O ID do cliente que está fazendo a solicitação.

principalId

O ID da entidade principal que está fazendo a solicitação.

protocolo

O protocolo usado ao criar a solicitação. Os valores válidos são `MQTT` ou `HTTP`.

sourceIp

O endereço IP no qual a solicitação foi originada.

sourcePort

A porta em que a solicitação foi originada.

disconnectReason

A razão pela qual o cliente está se desconectando.

Entrada de log GetRetainedMessage

OAWS IoTO agente de mensagens do gera uma entrada de log com um`eventType`do`GetRetainedMessage`quando`GetRetainedMessage`é chamado.

Exemplo de entrada de log GetRetainedMessage

```
{
    "timestamp": "2017-08-07 18:47:56.664",
    "logLevel": "INFO",
    "traceId": "1a60d02e-15b9-605b-7096-a9f584a6ad3f",
    "accountId": "123456789012",
    "status": "Success",
    "eventType": "GetRetainedMessage",
    "protocol": "HTTP",
    "topicName": "a/b/c",
    "qos": "1",
    "lastModifiedDate": "2017-08-07 18:47:56.664"
}
```

Além do Comum CloudWatch Atributos logs (p. 474), as entradas de log `GetRetainedMessage` contêm os seguintes atributos:

lastModifiedDate

A data e a hora da Époch, em milissegundos, quando a mensagem retida foi armazenada porAWS IoT.

protocolo

O protocolo usado ao criar a solicitação. Valor válido: `HTTP`.

qos

O nível de qualidade do serviço (QoS) usado na solicitação de publicação. Os valores válidos são `0` ou `1`.

topicName

O nome do tópico que você assinou.

Entrada de log ListRetainedMessage

OAWS IoTO agente de mensagens do gera uma entrada de log com um`eventType`do`ListRetainedMessage`quando`ListRetainedMessages`é chamado.

Exemplo de entrada de log ListRetainedMessage

```
{
    "timestamp": "2017-08-07 18:47:56.664",
    "logLevel": "INFO",
    "traceId": "1a60d02e-15b9-605b-7096-a9f584a6ad3f",
    "accountId": "123456789012",
    "status": "Success",
    "eventType": "ListRetainedMessage",
    "protocol": "HTTP"
}
```

Além doComum CloudWatch Atributos logs (p. 474),`ListRetainedMessage`as entradas de log contêm o seguinte atributo:

protocolo

O protocolo usado ao criar a solicitação. Valor válido: `HTTP`.

Entrada de log de publicação de entrada

Quando o agente de mensagens do AWS IoT recebe uma mensagem MQTT, ele gera uma entrada de log com um `eventType` de `Publish-In`.

Exemplo de entrada de log de publicação de entrada

```
{
        "timestamp": "2017-08-10 15:39:30.961",
        "logLevel": "INFO",
        "traceId": "672ec480-31ce-fd8b-b5fb-22e3ac420699",
        "accountId": "123456789012",
        "status": "Success",
        "eventType": "Publish-In",
        "protocol": "MQTT",
        "topicName": "$aws/things/MyThing/shadow/get",
        "clientId": "abf27092886e49a8a5c1922749736453",
        "principalId": "145179c40e2219e18a909d896a5340b74cf97a39641beec2fc3eeafc5a932167",
        "sourceIp": "205.251.233.181",
        "sourcePort": 13490,
        "retain": "True"
    }
```

Além do Comum CloudWatch Atributos logs (p. 474), as entradas de log `Publish-In` contêm os seguintes atributos:

clientId

O ID do cliente que está fazendo a solicitação.

principalId

O ID da entidade principal que está fazendo a solicitação.

protocolo

O protocolo usado ao criar a solicitação. Os valores válidos são `MQTT` ou `HTTP`.

retêm

O atributo usado quando uma mensagem tem o sinalizador RETAIN definido com um valor de`True`. Se a mensagem não tiver o sinalizador RETAIN definido, esse atributo não aparecerá na entrada de registro. Para obter mais informações, consulteUsar mensagens retidas do MQTT (p. 90)

sourceIp

O endereço IP no qual a solicitação foi originada.

sourcePort

A porta em que a solicitação foi originada.

topicName

O nome do tópico que você assinou.

Entrada de log de publicação de saída

Quando o agente de mensagens publica uma mensagem MQTT, ele gera um log com um `eventType` de `Publish-Out`.

Exemplo de entrada de log de publicação de saída

```
{
    "timestamp": "2017-08-10 15:39:30.961",
    "logLevel": "INFO",
    "traceId": "672ec480-31ce-fd8b-b5fb-22e3ac420699",
    "accountId": "123456789012",
    "status": "Success",
    "eventType": "Publish-Out",
    "protocol": "MQTT",
    "topicName": "$aws/things/MyThing/shadow/get",
    "clientId": "abf27092886e49a8a5c1922749736453",
    "principalId": "145179c40e2219e18a909d896a5340b74cf97a39641beec2fc3eeafc5a932167",
    "sourceIp": "205.251.233.181",
    "sourcePort": 13490
}
```

Além do Comum CloudWatch Atributos logs (p. 474), as entradas de log `Publish-Out` contêm os seguintes atributos:

clientId

O ID do cliente inscrito que recebe mensagens nesse tópico MQTT.

principalId

O ID da entidade principal que está fazendo a solicitação.

protocolo

O protocolo usado ao criar a solicitação. Os valores válidos são `MQTT` ou `HTTP`.

sourceIp

O endereço IP no qual a solicitação foi originada.

sourcePort

A porta em que a solicitação foi originada.

topicName

O nome do tópico que você assinou.

Entrada de log de assinatura

O agente de mensagens do AWS IoT gera um log com um `eventType` de `Subscribe` quando um cliente MQTT assina um tópico.

Exemplo de entrada de log de assinatura

```
{
    "timestamp": "2017-08-10 15:39:04.413",
    "logLevel": "INFO",
    "traceId": "7aa5c38d-1b49-3753-15dc-513ce4ab9fa6",
    "accountId": "123456789012",
    "status": "Success",
    "eventType": "Subscribe",
    "protocol": "MQTT",
    "topicName": "$aws/things/MyThing/shadow/#",
    "clientId": "abf27092886e49a8a5c1922749736453",
    "principalId": "145179c40e2219e18a909d896a5340b74cf97a39641beec2fc3eeafc5a932167",
    "sourceIp": "205.251.233.181",
    "sourcePort": 13490
}
```

Além do Comum CloudWatch Atributos logs (p. 474), as entradas de log `Subscribe` contêm os seguintes atributos:

clientId

O ID do cliente que está fazendo a solicitação.

principalId

O ID da entidade principal que está fazendo a solicitação.

protocolo

O protocolo usado ao criar a solicitação. Os valores válidos são `MQTT` ou `HTTP`.

sourceIp

O endereço IP no qual a solicitação foi originada.

sourcePort

A porta em que a solicitação foi originada.

topicName

O nome do tópico que você assinou.

Entradas de log do Device Shadow

O serviço Device Shadow do AWS IoT gera entradas de log para os seguintes eventos:

Tópicos

- Entrada de log DeleteThingShadow (p. 461)
- Entrada de log GetThingShadow (p. 461)
- Entrada de log UpdateThingShadow (p. 462)

Entrada de log DeleteThingShadow

O serviço Device Shadow gera uma entrada de log com um `eventType` de `DeleteThingShadow` quando uma solicitação para excluir a shadow de um dispositivo é recebida.

Exemplo de entrada de log DeleteThingShadow

```
{
    "timestamp": "2017-08-07 18:47:56.664",
    "logLevel": "INFO",
    "traceId": "1a60d02e-15b9-605b-7096-a9f584a6ad3f",
    "accountId": "123456789012",
    "status": "Success",
    "eventType": "DeleteThingShadow",
    "protocol": "MQTT",
    "deviceShadowName": "Jack",
    "topicName": "$aws/things/Jack/shadow/delete"
}
```

Além do Comum CloudWatch Atributos logs (p. 474), as entradas de log `DeleteThingShadow` contêm os seguintes atributos:

deviceShadowName

O nome da shadow a ser atualizada.

protocolo

O protocolo usado ao criar a solicitação. Os valores válidos são `MQTT` ou `HTTP`.

topicName

O nome do tópico no qual a solicitação foi publicada.

Entrada de log GetThingShadow

O serviço Device Shadow gera uma entrada de log com um `eventType` de `GetThingShadow` quando uma solicitação para obter uma shadow é recebida.

Exemplo de entrada de log GetThingShadow

```
{
    "timestamp": "2017-08-09 17:56:30.941",
    "logLevel": "INFO",
    "traceId": "b575f19a-97a2-cf72-0ed0-c64a783a2504",
    "accountId": "123456789012",
    "status": "Success",
    "eventType": "GetThingShadow",
    "protocol": "MQTT",
    "deviceShadowName": "MyThing",
```

```
    "topicName": "$aws/things/MyThing/shadow/get"
}
```

Além do Comum CloudWatch Atributos logs (p. 474), as entradas de log `GetThingShadow` contêm os seguintes atributos:

deviceShadowName

O nome da shadow solicitada.

protocolo

O protocolo usado ao criar a solicitação. Os valores válidos são `MQTT` ou `HTTP`.

topicName

O nome do tópico no qual a solicitação foi publicada.

Entrada de log UpdateThingShadow

O serviço Device Shadow gera uma entrada de log com um `eventType` de `UpdateThingShadow` quando uma solicitação para atualizar a shadow de um dispositivo é recebida.

Exemplo de entrada de log UpdateThingShadow

```
{
    "timestamp": "2017-08-07 18:43:59.436",
    "logLevel": "INFO",
    "traceId": "d0074ba8-0c4b-a400-69df-76326d414c28",
    "accountId": "123456789012",
    "status": "Success",
    "eventType": "UpdateThingShadow",
    "protocol": "MQTT",
    "deviceShadowName": "Jack",
    "topicName": "$aws/things/Jack/shadow/update"
}
```

Além do Comum CloudWatch Atributos logs (p. 474), as entradas de log `UpdateThingShadow` contêm os seguintes atributos:

deviceShadowName

O nome da shadow a ser atualizada.

protocolo

O protocolo usado ao criar a solicitação. Os valores válidos são `MQTT` ou `HTTP`.

topicName

O nome do tópico no qual a solicitação foi publicada.

Entradas de log do mecanismo de regras

O mecanismo de regras do AWS IoT Rules gera logs para os seguintes eventos:

Tópicos

- Entrada de log FunctionExecution (p. 463)
- Entrada de log RuleExecution (p. 463)
- Entrada de log RuleMatch (p. 464)

- Entrada de log RuleMessageThrottled (p. 465)
- Entrada de log RuleNotFound (p. 466)
- Entrada de log StartingRuleExecution (p. 466)

Entrada de log FunctionExecution

O mecanismo de regras gera uma entrada de log com um `eventType` de `FunctionExecution` quando uma consulta SQL de uma regra chama uma função externa. Uma função externa é chamada quando uma ação de regra faz uma solicitação HTTP para a AWS IoT ou para outro serviço web (por exemplo, chamar `get_thing_shadow` ou `machinelearning_predict`).

Exemplo de entrada de log FunctionExecution

```
{
    "timestamp": "2017-07-13 18:33:51.903",
    "logLevel": "DEBUG",
    "traceId": "180532b7-0cc7-057b-687a-5ca1824838f5",
    "status": "Success",
    "eventType": "FunctionExecution",
    "clientId": "N/A",
    "topicName":"rules/test",
    "ruleName": "ruleTestPredict",
    "ruleAction": "MachinelearningPredict",
    "resources": {
        "ModelId": "predict-model"
    },
    "principalId": "145179c40e2219e18a909d896a5340b74cf97a39641beec2fc3eeafc5a932167"
}
```

Além do Comum CloudWatch Atributos logs (p. 474), as entradas de log `FunctionExecution` contêm os seguintes atributos:

clientId

`N/A` para logs `FunctionExecution`.

principalId

O ID da entidade principal que está fazendo a solicitação.

recursos

Uma coleção de recursos usados pelas ações da regra.

ruleName

O nome da regra correspondente.

topicName

O nome do tópico que você assinou.

Entrada de log RuleExecution

Quando o mecanismo de regras do AWS IoT aciona uma ação de regra, ele gera uma entrada de log `RuleExecution`.

Exemplo de entrada de log RuleExecution

```
{
    "timestamp": "2017-08-10 16:32:46.070",
```

```
    "logLevel": "INFO",
    "traceId": "30aa7ccc-1d23-0b97-aa7b-76196d83537e",
    "accountId": "123456789012",
    "status": "Success",
    "eventType": "RuleExecution",
    "clientId": "abf27092886e49a8a5c1922749736453",
    "topicName": "rules/test",
    "ruleName": "JSONLogsRule",
    "ruleAction": "RepublishAction",
    "resources": {
        "RepublishTopic": "rules/republish"
    },
    "principalId": "145179c40e2219e18a909d896a5340b74cf97a39641beec2fc3eeafc5a932167"
}
```

Além do Comum CloudWatch Atributos logs (p. 474), as entradas de log `RuleExecution` contêm os seguintes atributos:

clientId

O ID do cliente que está fazendo a solicitação.

principalId

O ID da entidade principal que está fazendo a solicitação.

recursos

Uma coleção de recursos usados pelas ações da regra.

ruleAction

O nome da ação disparada.

ruleName

O nome da regra correspondente.

topicName

O nome do tópico que você assinou.

Entrada de log RuleMatch

O mecanismo de regras do AWS IoT gera uma entrada de log com um `eventType` de `RuleMatch` quando o agente de mensagens recebe uma mensagem que corresponde a uma regra.

Exemplo de entrada de log RuleMatch

```
{
    "timestamp": "2017-08-10 16:32:46.002",
    "logLevel": "INFO",
    "traceId": "30aa7ccc-1d23-0b97-aa7b-76196d83537e",
    "accountId": "123456789012",
    "status": "Success",
    "eventType": "RuleMatch",
    "clientId": "abf27092886e49a8a5c1922749736453",
    "topicName": "rules/test",
    "ruleName": "JSONLogsRule",
    "principalId": "145179c40e2219e18a909d896a5340b74cf97a39641beec2fc3eeafc5a932167"
}
```

Além do Comum CloudWatch Atributos logs (p. 474), as entradas de log `RuleMatch` contêm os seguintes atributos:

clientId

O ID do cliente que está fazendo a solicitação.

principalId

O ID da entidade principal que está fazendo a solicitação.

ruleName

O nome da regra correspondente.

topicName

O nome do tópico que você assinou.

Entrada de log RuleMessageThrottled

Quando uma mensagem é limitada, o mecanismo de regras do AWS IoT gera uma entrada de log com um `eventType` de `RuleMessageThrottled`.

Exemplo de entrada de log RuleMessageThrottled

```
{
    "timestamp": "2017-10-04 19:25:46.070",
    "logLevel": "ERROR",
    "traceId": "30aa7ccc-1d23-0b97-aa7b-76196d83537e",
    "accountId": "123456789012",
    "status": "Failure",
    "eventType": "RuleMessageThrottled",
    "clientId": "abf27092886e49a8a5c1922749736453",
    "topicName": "$aws/rules/example_rule",
    "ruleName": "example_rule",
    "principalId": "145179c40e2219e18a909d896a5340b74cf97a39641beec2fc3eeafc5a932167",
    "reason": "RuleExecutionThrottled",
    "details": "Message for Rule example_rule throttled"
}
```

Além do Comum CloudWatch Atributos logs (p. 474), as entradas de log `RuleMessageThrottled` contêm os seguintes atributos:

clientId

O ID do cliente que está fazendo a solicitação.

detalhes

Uma breve explicação do erro.

principalId

O ID da entidade principal que está fazendo a solicitação.

reason

A string "RuleMessageThrottled".

ruleName

O nome da regra a ser acionada.

topicName

O nome do tópico publicado.

Entrada de log RuleNotFound

Quando o mecanismo de regras do AWS IoT não consegue encontrar uma regra com um nome específico, ele gera uma entrada de log com um `eventType` de `RuleNotFound`.

Exemplo de entrada de log RuleNotFound

```
{
    "timestamp": "2017-10-04 19:25:46.070",
    "logLevel": "ERROR",
    "traceId": "30aa7ccc-1d23-0b97-aa7b-76196d83537e",
    "accountId": "123456789012",
    "status": "Failure",
    "eventType": "RuleNotFound",
    "clientId": "abf27092886e49a8a5c1922749736453",
    "topicName": "$aws/rules/example_rule",
    "ruleName": "example_rule",
    "principalId": "145179c40e2219e18a909d896a5340b74cf97a39641beec2fc3eeafc5a932167",
    "reason": "RuleNotFound",
    "details": "Rule example_rule not found"
}
```

Além do Comum CloudWatch Atributos logs (p. 474), as entradas de log `RuleNotFound` contêm os seguintes atributos:

clientId

O ID do cliente que está fazendo a solicitação.

detalhes

Uma breve explicação do erro.

principalId

O ID da entidade principal que está fazendo a solicitação.

reason

A string "RuleNotFound".

ruleName

O nome da regra que não pôde ser encontrada.

topicName

O nome do tópico publicado.

Entrada de log StartingRuleExecution

Quando o mecanismo de regras do AWS IoT aciona uma ação de regra, ele gera uma entrada de log com um `eventType` de `StartingRuleExecution`.

Exemplo de entrada de log StartingRuleExecution

```
{
    "timestamp": "2017-08-10 16:32:46.002",
    "logLevel": "DEBUG",
    "traceId": "30aa7ccc-1d23-0b97-aa7b-76196d83537e",
    "accountId": "123456789012",
    "status": "Success",
    "eventType": "StartingRuleExecution",
```

```
    "clientId": "abf27092886e49a8a5c1922749736453",
    "topicName": "rules/test",
    "ruleName": "JSONLogsRule",
    "ruleAction": "RepublishAction",
    "principalId": "145179c40e2219e18a909d896a5340b74cf97a39641beec2fc3eeafc5a932167"
}
```

Além do Comum CloudWatch Atributos logs (p. 474), as entradas de log `rule-` contêm os seguintes atributos:

clientId

O ID do cliente que está fazendo a solicitação.

principalId

O ID da entidade principal que está fazendo a solicitação.

ruleAction

O nome da ação disparada.

ruleName

O nome da regra correspondente.

topicName

O nome do tópico que você assinou.

Entradas de log de trabalhos

O serviço de trabalho do AWS IoT gera entradas de log para os eventos a seguir. As entradas de log são geradas quando uma solicitação HTTP ou MQTT é recebida do dispositivo.

Tópicos

- Entrada de log DescribeJobExecution (p. 467)
- Entrada de log GetPendingJobExecution (p. 468)
- Entrada de log ReportFinalJobExecutionCount (p. 469)
- Entrada de log StartNextPendingJobExecution (p. 469)
- Entrada de log UpdateJobExecution (p. 470)

Entrada de log DescribeJobExecution

O serviço de trabalhos do AWS IoT gera uma entrada de log com um `eventType` de `DescribeJobExecution` quando o serviço recebe uma solicitação para descrever a execução de um trabalho.

Exemplo de entrada de log DescribeJobExecution

```
{
    "timestamp": "2017-08-10 19:13:22.841",
    "logLevel": "DEBUG",
    "accountId": "123456789012",
    "status": "Success",
    "eventType": "DescribeJobExecution",
    "protocol": "MQTT",
    "clientId": "thingOne",
    "jobId": "002",
    "topicName": "$aws/things/thingOne/jobs/002/get",
```

```
    "clientToken": "myToken",
    "details": "The request status is SUCCESS."
}
```

Além do Comum CloudWatch Atributos logs (p. 474), as entradas de log `GetJobExecution` contêm os seguintes atributos:

clientId

O ID do cliente que está fazendo a solicitação.

clientToken

Um identificador exclusivo e que diferencia maiúsculas e minúsculas, para garantir a idempotência da solicitação. Para obter mais informações, consulte Como garantir idempotência.

detalhes

Outras informações do serviço Jobs.

jobId

O ID do trabalho para a execução do trabalho.

protocolo

O protocolo usado ao criar a solicitação. Os valores válidos são `MQTT` ou `HTTP`.

topicName

O tópico usado para fazer a solicitação.

Entrada de log GetPendingJobExecution

O serviço de trabalhos do AWS IoT gera uma entrada de log com um `eventType` de `GetPendingJobExecution` quando o serviço recebe uma solicitação para a execução de um trabalho.

Exemplo de entrada de log GetPendingJobExecution

```
{
    "timestamp": "2018-06-13 17:45:17.197",
    "logLevel": "DEBUG",
    "accountId": "123456789012",
    "status": "Success",
    "eventType": "GetPendingJobExecution",
    "protocol": "MQTT",
    "clientId": "299966ad-54de-40b4-99d3-4fc8b52da0c5",
    "topicName": "$aws/things/299966ad-54de-40b4-99d3-4fc8b52da0c5/jobs/get",
    "clientToken": "24b9a741-15a7-44fc-bd3c-1ff2e34e5e82",
    "details": "The request status is SUCCESS."
}
```

Além do Comum CloudWatch Atributos logs (p. 474), as entradas de log `GetPendingJobExecution` contêm os seguintes atributos:

clientId

O ID do cliente que está fazendo a solicitação.

clientToken

Um identificador exclusivo e que diferencia maiúsculas e minúsculas, para garantir a idempotência da solicitação. Para obter mais informações, consulte Como garantir idempotência.

detalhes

Outras informações do serviço Jobs.

protocolo

O protocolo usado ao criar a solicitação. Os valores válidos são `MQTT` ou `HTTP`.

topicName

O nome do tópico que você assinou.

Entrada de log ReportFinalJobExecutionCount

OAWS IoTO serviço de trabalho do gera uma entrada de log com um`entryType`do`ReportFinalJobExecutionCount`Quando um trabalho é concluído.

Exemplo de entrada de log ReportFinalJobExecutionCount

```
{
    "timestamp": "2017-08-10 19:44:16.776",
    "logLevel": "INFO",
    "accountId": "123456789012",
    "status": "Success",
    "eventType": "ReportFinalJobExecutionCount",
    "jobId": "002",
    "details": "Job 002 completed. QUEUED job execution count: 0 IN_PROGRESS job execution
 count: 0 FAILED job execution count: 0 SUCCEEDED job execution count: 1 CANCELED job
 execution count: 0 REJECTED job execution count: 0 REMOVED job execution count: 0"
}
```

Além do Comum CloudWatch Atributos logs (p. 474), as entradas de log `ReportFinalJobExecutionCount` contêm os seguintes atributos:

detalhes

Outras informações do serviço Jobs.

jobId

O ID do trabalho para a execução do trabalho.

Entrada de log StartNextPendingJobExecution

Quando ele recebe uma solicitação para iniciar a execução do próximo trabalho pendente, o serviço de trabalhos do AWS IoT gera uma entrada de log com um `eventType` de `StartNextPendingJobExecution`.

Exemplo de entrada de log StartNextPendingJobExecution

```
{
    "timestamp": "2018-06-13 17:49:51.036",
    "logLevel": "DEBUG",
    "accountId": "123456789012",
    "status": "Success",
    "eventType": "StartNextPendingJobExecution",
    "protocol": "MQTT",
    "clientId": "95c47808-b1ca-4794-bc68-a588d6d9216c",
    "topicName": "$aws/things/95c47808-b1ca-4794-bc68-a588d6d9216c/jobs/start-next",
    "clientToken": "bd7447c4-3a05-49f4-8517-dd89b2c68d94",
    "details": "The request status is SUCCESS."
```

```
}
```

Além do Comum CloudWatch Atributos logs (p. 474), as entradas de log `StartNextPendingJobExecution` contêm os seguintes atributos:

clientId

O ID do cliente que está fazendo a solicitação.

clientToken

Um identificador exclusivo e que diferencia maiúsculas e minúsculas, para garantir a idempotência da solicitação. Para obter mais informações, consulte Como garantir idempotência.

detalhes

Outras informações do serviço Jobs.

protocolo

O protocolo usado ao criar a solicitação. Os valores válidos são `MQTT` ou `HTTP`.

topicName

O tópico usado para fazer a solicitação.

Entrada de log UpdateJobExecution

O serviço de trabalhos do AWS IoT gera uma entrada de log com um `eventType` de `UpdateJobExecution` quando o serviço recebe uma solicitação para atualizar a execução de um trabalho.

Exemplo de entrada de log UpdateJobExecution

```
{
    "timestamp": "2017-08-10 19:25:14.758",
    "logLevel": "DEBUG",
    "accountId": "123456789012",
    "status": "Success",
    "eventType": "UpdateJobExecution",
    "protocol": "MQTT",
    "clientId": "thingOne",
    "jobId": "002",
    "topicName": "$aws/things/thingOne/jobs/002/update",
    "clientToken": "myClientToken",
    "versionNumber": "1",
    "details": "The destination status is IN_PROGRESS. The request status is SUCCESS."
}
```

Além do Comum CloudWatch Atributos logs (p. 474), as entradas de log `UpdateJobExecution` contêm os seguintes atributos:

clientId

O ID do cliente que está fazendo a solicitação.

clientToken

Um identificador exclusivo e que diferencia maiúsculas e minúsculas, para garantir a idempotência da solicitação. Para obter mais informações, consulte Como garantir idempotência.

detalhes

Outras informações do serviço Jobs.

jobId

O ID do trabalho para a execução do trabalho.

protocolo

O protocolo usado ao criar a solicitação. Os valores válidos são `MQTT` ou `HTTP`.

topicName

O tópico usado para fazer a solicitação.

versionNumber

A versão da execução do trabalho.

Entradas de log de provisionamento de dispositivos

O serviço de provisionamento de dispositivos da AWS IoT gera logs para os seguintes eventos.

Tópicos

- Entrada de log GetDeviceCredentials (p. 471)
- Entrada de log ProvisionDevice (p. 471)

Entrada de log GetDeviceCredentials

O serviço de provisionamento de dispositivos do AWS IoT gera uma entrada de log com um `eventType` de `GetDeviceCredential` quando um cliente chama `GetDeviceCredential`.

Exemplo de entrada de log GetDeviceCredentials

```
{
  "timestamp" : "2019-02-20 20:31:22.932",
  "logLevel" : "INFO",
  "traceId" : "8d9c016f-6cc7-441e-8909-7ee3d5563405",
  "accountId" : "123456789101",
  "status" : "Success",
  "eventType" : "GetDeviceCredentials",
  "deviceCertificateId" :
 "e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855",
  "details" : "Additional details about this log."
}
```

Além do Comum CloudWatch Atributos logs (p. 474), as entradas de log `GetDeviceCredentials` contêm os seguintes atributos:

detalhes

Uma breve explicação do erro.

deviceCertificateId

O ID do certificado do dispositivo.

Entrada de log ProvisionDevice

O serviço de provisionamento de dispositivos do AWS IoT gera uma entrada de log com um `eventType` de `ProvisionDevice` quando um cliente chama `ProvisionDevice`.

Exemplo de entrada de log ProvisionDevice

```
{
  "timestamp" : "2019-02-20 20:31:22.932",
  "logLevel" : "INFO",
  "traceId" : "8d9c016f-6cc7-441e-8909-7ee3d5563405",
  "accountId" : "123456789101",
  "status" : "Success",
  "eventType" : "ProvisionDevice",
  "provisioningTemplateName" : "myTemplate",
  "deviceCertificateId" :
 "e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855",
  "details" : "Additional details about this log."
 }
```

Além do Comum CloudWatch Atributos logs (p. 474), as entradas de log `ProvisionDevice` contêm os seguintes atributos:

detalhes

Uma breve explicação do erro.

deviceCertificateId

O ID do certificado do dispositivo.

provisioningTemplateName

O nome do modelo de provisionamento.

Entradas de log de grupo de coisas dinâmico

Grupos de coisas dinâmicas da AWS IoT geram registros para o seguinte evento.

Tópicos

- Entrada de log AddThingToDynamicThingGroupsFailed (p. 472)

Entrada de log AddThingToDynamicThingGroupsFailed

Quando a AWS IoT não consegue adicionar uma coisa aos grupos dinâmicos especificados, ela gera uma entrada de log com um `eventType` de `AddThingToDynamicThingGroupsFailed`. Isso acontece quando uma coisa atendia aos critérios para estar no grupo dinâmico e, no entanto, não pôde ser adicionada ao grupo dinâmico ou foi removida do grupo dinâmico. Isso pode acontecer porque:

- A coisa já pertence ao número máximo de grupos.
- A opção --override-dynamic-groups foi usada para adicionar a coisa a um grupo de coisas estáticas. Foi removido de um grupo dinâmico para tornar isso possível.

Para obter mais informações, consulte Limitações e conflitos de grupo de coisas dinâmico (p. 285).

Exemplo de entrada de log AddThingToDynamicThingGroupsFailed

Este exemplo mostra a entrada de log de um erro `AddThingToDynamicThingGroupsFailed`. Neste exemplo, TestThing atendeu aos critérios para estar nos grupos de coisas dinâmicas listados em `dynamicThingGroupNames`, mas não foi adicionado a esses grupos dinâmicos, conforme descrito em `reason`.

```
{
```

```
 "timestamp": "2020-03-16 22:24:43.804",
 "logLevel": "ERROR",
 "traceId": "70b1f2f5-d95e-f897-9dcc-31e68c3e1a30",
 "accountId": "57EXAMPLE833",
 "status": "Failure",
 "eventType": "AddThingToDynamicThingGroupsFailed",
 "thingName": "TestThing",
 "dynamicThingGroupNames": [
  "DynamicThingGroup11",
  "DynamicThingGroup12",
  "DynamicThingGroup13",
  "DynamicThingGroup14"
 ],
 "reason": "The thing failed to be added to the given dynamic thing group(s) because the
 thing already belongs to the maximum allowed number of groups."
}
```

Além do Comum CloudWatch Atributos logs (p. 474), as entradas de log `AddThingToDynamicThingGroupsFailed` contêm os seguintes atributos:

dynamicThingGroupNames

Uma matriz dos grupos de coisas dinâmicas aos quais a coisa não foi adicionada.

reason

A razão pela qual a coisa não foi adicionada aos grupos de coisas dinâmicas.

thingName

O nome da coisa que não foi adicionada a um grupo dinâmico.

Entradas de log de indexação de fro

AWS IoTA indexação de frotas do gera logs para os eventos a seguir.

Tópicos

- Entrada de log DeviceDefenderThingViolationSeventSizelimitExceeded (p. 473)
- Entrada de log NamedShadoweventSizLimitExceeded (p. 474)
- Entrada de log NamedShadowCountForDynamicGroupQueryLimitExceeded (p. 474)

Entrada de log DeviceDefenderThingViolationSeventSizelimitExceeded

O tamanho total dos dados de uma coisa processada pela indexação da frota é limitado a 32 KB. Quando esse limite for violado por uma coisa devido a um evento de violações do Device Defender, o`DeviceDefenderThingViolationsEventSizeLimitExceeded`o tipo de evento será emitido.

Exemplo de entrada de log DeviceDefenderThingViolationSeventSizelimitExceeded

Este exemplo mostra a entrada de log de um`DeviceDefenderThingViolationsEventSizeLimitExceeded`(Nenhum console do Windows encontrado. Você está executando o cmd.exe?”. Neste exemplo, a coisa chamada`TestThing`tem dados de violações que estavam sendo preenchidos, mas a indexação ou a percolação dinâmica do grupo de eventos de violação foi ignorada, conforme descrito na`reason`campo.

```
{
"timestamp": "2020-03-16 22:24:43.804",
"logLevel": "ERROR",
"traceId": "70b1f2f5-d95e-f897-9dcc-31e68c3e1a30",
"accountId": "571032923833",
```

```
"status": "Failure",
"eventType": "DeviceDefenderThingViolationsEventSizeLimitExceeded",
"thingName": "TestThing",
"reason": "Device Defender Thing Violations event skipped because 32 KB size limit was
 exceeded."
}
```

Entrada de log NamedShadoweventSizLimitExceeded

O tamanho total dos dados de uma coisa processada pela indexação da frota é limitado a 32 KB. Quando esse limite é violado por uma coisa devido a um evento sombra nomeado, o`NamedShadowEventSizeLimitExceeded`o tipo de evento será emitido.

Exemplo de entrada de log NamedShadoweventSizLimitExceeded

Este exemplo mostra a entrada de log de um`NamedShadowEventSizeLimitExceeded`(Nenhum console do Windows encontrado. Você está executando o cmd.exe?". Neste exemplo, a sombra nomeada (`myTestNamedShadow`) dados da coisa (`TestThing`) estava sendo preenchido, mas a indexação ou percolação de grupo dinâmico de`NamedShadow`evento foi ignorado, conforme descrito na`reason`campo.

```
{
"timestamp": "2020-03-16 22:24:43.804",
"logLevel": "ERROR",
"traceId": "70b1f2f5-d95e-f897-9dcc-31e68c3e1a30",
"accountId": "571032923833",
"status": "Failure",
"eventType": "NamedShadowEventSizeLimitExceeded",
"thingName": "TestThing",
"namedShadowName": "myTestNamedShadow",
"reason": "Named shadow event skipped because 32 KB size limit was exceeded."
}
```

Entrada de log NamedShadowCountForDynamicGroupQueryLimitExceeded

No máximo 5 sombras nomeadas por coisa são processadas para termos de consulta que não são específicos da fonte de dados em grupos dinâmicos. Quando esse limite é violado por uma coisa, o`NamedShadowCountForDynamicGroupQueryLimitExceeded`o tipo de evento será emitido.

Exemplo de entrada de log NamedShadowCountForDynamicGroupQueryLimitExceeded

Este exemplo mostra a entrada de log de um`NamedShadowCountForDynamicGroupQueryLimitExceeded`(Nenhum console do Windows encontrado. Você está executando o cmd.exe?". Neste exemplo, com base em todos os valores`DynamicGroup`os resultados podem ser imprecisos, conforme descrito no`reason`campo.

```
{
"timestamp": "2020-03-16 22:24:43.804",
"logLevel": "ERROR",
"traceId": "70b1f2f5-d95e-f897-9dcc-31e68c3e1a30",
"accountId": "571032923833",
"status": "Failure",
"eventType": "NamedShadowCountForDynamicGroupQueryLimitExceeded",
"thingName": "TestThing",
"reason": "A maximum of 5 named shadows per thing are processed for non-data source
 specific query terms in dynamic groups."
}
```

Comum CloudWatch Atributos logs

Tudo CloudWatch As entradas de log de logs do incluem estes atributos:

accountId

SuasConta da AWSID.

eventType

O tipo de evento para o qual o log foi gerado. O valor do tipo de evento depende do evento que gerou a entrada de log. Cada descrição de entrada de log inclui o valor de `eventType` para essa entrada de log.

logLevel

O nível de log que está sendo usado. Para obter mais informações, consultethe section called “Níveis de log” (p. 438)

status

O status da solicitação.

timestamp

A data e hora do UNIX de quando o cliente se conectou ao agente de mensagens da AWS IoT.

traceId

Um identificador gerado aleatoriamente que pode ser usado para correlacionar todos os logs para uma solicitação específica.

Registre em log as chamadas à API do AWS IoT usando o AWS CloudTrail.

AWS IoTO é integrado aoAWS CloudTrail, um serviço que fornece um registro de ações executadas por um usuário, uma função ou umAWSserviço noAWS IoT. CloudTrail O captura todas as chamadas à API doAWS IoTComo eventos, incluindo chamadas doAWS IoTConsole e as chamadas de código para oAWS IoTAPIs. 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. Se não configurar uma trilha, você ainda poderá visualizar os eventos mais recentes na CloudTrail Console noHistórico do evento. Usando as informações coletadas pela CloudTrail, é possível determinar a solicitação que foi feita aoAWS IoT, O endereço IP do qual a solicitação foi feita, quem fez a solicitação, quando ela foi feita, e outros detalhes.

Para saber mais sobre o CloudTrail, consulte o Guia do usuário do AWS CloudTrail.

Informações do AWS IoT no CloudTrail

O CloudTrail é habilitado em sua Conta da AWS quando ela é criada. Quando a atividade ocorre emAWS IoT, essa atividade é registrada em um CloudTrail evento junto com outrosAWSEventos de serviço noHistórico do evento. Você pode visualizar, pesquisar e baixar eventos recentes em sua Conta da AWS. Para obter mais informações, consulteVisualizar eventos do com CloudTrail Histórico do evento.

Para obter um registro contínuo de eventos na sua Conta da AWS, incluindo eventos para o AWS IoT, crie uma trilha. Uma trilha permite CloudTrail Para fornecer arquivos de log a um bucket do Amazon S3. Por padrão, quando você cria uma trilha no console, ela é aplicada a todas as Região da AWS. A trilha registra em log eventos em cada Região da AWS na partição da AWS e entrega os arquivos de log para o bucket do Amazon S3 especificado por você. Você pode configurar outrosAWSServiços para analisar e atuar mais profundamente sobre os dados de eventos coletados em CloudTrail logs. 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
- Recebendo CloudTrail Arquivos de log de várias regiõeseRecebimento de arquivos de log do CloudTrail de várias contas

> **Note**
>
> AWS IoTAs ações do plano de dados (lado do dispositivo) não são registradas em log pelo CloudTrail. Usar o CloudWatch Para monitorar essas ações.

De um modo geral,AWS IoTAs ações do plano de controle que fazem alterações são registradas em log pelo CloudTrail. Chamadas comoCreateThing,CreateKeysAndCertificate, eUpdateCertificatesair CloudTrail entradas, enquanto chamadas comoListThingseListTopicRulesNão.

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 raiz ou do IAM.
- 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 o Elemento userIdentity do CloudTrail.

AWS IoTAs ações do são documentadas noAWS IoTReferência de API do.AWS IoT As ações sem fio são documentadas noAWS IoTReferência de API sem fio.

Noções básicas sobre entradas de arquivos de log do AWS IoT

Uma trilha é uma configuração que permite a entrega de eventos como arquivos de log a um bucket do Amazon S3 especificado. CloudTrail Os arquivos de log do contêm uma ou mais entradas de log. Um evento representa uma única solicitação de qualquer origem e inclui informações sobre a ação solicitada, a data e a hora da ação, os parâmetros de solicitação e assim por diante. CloudTrail Os arquivos de log do 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 um CloudTrail Entrada de log do que demonstra a`AttachPolicy`Ação .

```
{
    "timestamp":"1460159496",
    "AdditionalEventData":"",
    "Annotation":"",
    "ApiVersion":"",
    "ErrorCode":"",
    "ErrorMessage":"",
    "EventID":"8bff4fed-c229-4d2d-8264-4ab28a487505",
    "EventName":"AttachPolicy",
    "EventTime":"2016-04-08T23:51:36Z",
    "EventType":"AwsApiCall",
    "ReadOnly":"",
    "RecipientAccountList":"",
    "RequestID":"d4875df2-fde4-11e5-b829-23bf9b56cbcd",
    "RequestParamters":{
        "principal":"arn:aws:iot:us-
east-1:123456789012:cert/528ce36e8047f6a75ee51ab7beddb4eb268ad41d2ea881a10b67e8e76924d894",
        "policyName":"ExamplePolicyForIoT"
```

```
    },
    "Resources":"",
    "ResponseElements":"",
    "SourceIpAddress":"52.90.213.26",
    "UserAgent":"aws-internal/3",
    "UserIdentity":{
        "type":"AssumedRole",
        "principalId":"AKIAI44QH8DHBEXAMPLE",
        "arn":"arn:aws:sts::12345678912:assumed-role/iotmonitor-us-east-1-beta-
InstanceRole-1C5T1YCYMHPYT/i-35d0a4b6",
        "accountId":"222222222222",
        "accessKeyId":"access-key-id",
        "sessionContext":{
            "attributes":{
                "mfaAuthenticated":"false",
                "creationDate":"Fri Apr 08 23:51:10 UTC 2016"
            },
            "sessionIssuer":{
                "type":"Role",
                "principalId":"AKIAI44QH8DHBEXAMPLE",
                "arn":"arn:aws:iam::123456789012:role/executionServiceEC2Role/iotmonitor-
us-east-1-beta-InstanceRole-1C5T1YCYMHPYT",
                "accountId":"222222222222",
                "userName":"iotmonitor-us-east-1-InstanceRole-1C5T1YCYMHPYT"
            }
        },
        "invokedBy":{
            "serviceAccountId":"111111111111"
        }
    },
    "VpcEndpointId":""
}
```

Regras para AWS IoT

As regras permitem que os dispositivos interajam com os serviços da AWS. As regras são analisadas e as ações são realizadas com base no fluxo de tópicos do MQTT. Você pode usar regras para oferecer suporte a tarefas como estas:

- Aumentar ou filtrar os dados recebidos de um dispositivo.
- Gravar os dados recebidos de um dispositivo em um banco de dados do Amazon DynamoDB.
- Salve um arquivo no Amazon S3.
- Enviar uma notificação por push para todos os usuários usando o Amazon SNS.
- Publicar dados em uma fila do Amazon SQS.
- Invocar uma função do Lambda para extrair dados.
- Processar mensagens de um grande número de dispositivos usando o Amazon Kinesis.
- Enviar dados para a Amazon OpenSearch Serviço.
- Captura de um CloudWatch Métrica do.
- Altere um CloudWatch Alarme do
- Enviar os dados de uma mensagem MQTT para o Amazon Machine Learning para fazer previsões com base em um modelo do Amazon ML.
- Enviar uma mensagem para o fluxo de entrada da Salesforce IoT
- Envie dados de mensagem para um canal do AWS IoT Analytics.
- Iniciar a execução de uma máquina de estado do Step Functions.
- Envie dados de mensagem para uma entrada do AWS IoT Events.
- Envie dados de mensagem para uma propriedade de ativo em AWS IoT SiteWise.
- Envie dados de mensagens para um serviço ou aplicativo web.

Suas regras podem usar mensagens MQTT que passam pelo protocolo de publicação/assinatura suportado pelothe section called “Protocolos de comunicação de” (p. 83)ou, usando oIngestão Básica (p. 567)recurso, você pode enviar dados do dispositivo com segurança para oAWSserviços listados acima sem incorrerCustos de mensagens. (OIngestão Básica (p. 567)O recurso otimiza o fluxo de dados removendo o agente da mensagem de publicação/assinatura do caminho de ingestão, por isso é mais econômico, além de manter os recursos de segurança e processamento de dadosAWS IoT.)

AntesAWS IoTPode executar essas ações, você deve conceder a ele a permissão para acessar oAWSRecursos em seu nome. Quando as ações forem realizadas, você será cobrado pelas taxas padrão dos serviços da AWS utilizados.

Índice

- Concessão de umAWS IoTgoverne o acesso necessário (p. 479)
- Transmitir permissões de função (p. 480)
- Criar uma regra do AWS IoT (p. 481)
- Visualizar as regras (p. 485)
- Excluir uma regra (p. 485)
- Ações de regra do AWS IoT (p. 485)
- Solucionar problemas de uma regra (p. 559)
- Como acessar recursos entre contas usandoAWS IoTregras (p. 559)
- Tratamento de erros (ação de erro) (p. 565)
- Reduzir custos do sistema de mensagens com a Ingestão básica (p. 567)

- Referência SQL do AWS IoT (p. 568)

Concessão de umAWS IoTgoverne o acesso necessário

Você usa funções do IAM para controlar oAWSRecursos aos quais cada regra tem acesso. Antes de criar uma regra, você deve criar uma função do IAM com uma política que permita o acesso ao necessárioAWSrecursos da AWS.AWS IoTO assume essa função ao executar uma regra.

Para criar a função do IAM eAWS IoTpolítica que concede umAWS IoTgoverne o acesso necessário (AWS CLI)

1. Salve o seguinte documento de política de confiança, que concedeAWS IoTpermissão para assumir a função, para um arquivo chamado`iot-role-trust.json`.

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

   Use o comando create-role para criar uma função do IAM especificando o arquivo `iot-role-trust.json`:

```
aws iam create-role --role-name my-iot-role --assume-role-policy-document file://iot-role-trust.json
```

   A saída desse comando é semelhante à seguinte:

```
{
  "Role": {
    "AssumeRolePolicyDocument": "url-encoded-json",
    "RoleId": "AKIAIOSFODNN7EXAMPLE",
    "CreateDate": "2015-09-30T18:43:32.821Z",
    "RoleName": "my-iot-role",
    "Path": "/",
    "Arn": "arn:aws:iam::123456789012:role/my-iot-role"
  }
}
```

2. Salve o seguinte JSON em um arquivo chamado `my-iot-policy.json`.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": "dynamodb:*",
    "Resource": "*"
  }]
}
```

Esse JSON é um exemplo de documento de política que concedeAWS IoTO acesso de administrador ao DynamoDB.

Use o comando create-policy para conceder à AWS IoT o acesso aos seus recursos da AWS após assumir a função, transmitindo o arquivo `my-iot-policy.json`:

```
aws iam create-policy --policy-name my-iot-policy --policy-document file://my-iot-policy.json
```

Para obter mais informações sobre como conceder acesso aos serviços da AWS em políticas para AWS IoT, consulte Criar uma regra do AWS IoT (p. 481).

A saída do comando create-policy contém o ARN da política. Você precisa anexar a política a uma função.

```
{
  "Policy": {
    "PolicyName": "my-iot-policy",
    "CreateDate": "2015-09-30T19:31:18.620Z",
    "AttachmentCount": 0,
    "IsAttachable": true,
    "PolicyId": "ZXR6A36LTYANPAI7NJ5UV",
    "DefaultVersionId": "v1",
    "Path": "/",
    "Arn": "arn:aws:iam::123456789012:policy/my-iot-policy",
    "UpdateDate": "2015-09-30T19:31:18.620Z"
  }
}
```

3. Use o comando attach-role-policy para anexar a política à sua função:

```
aws iam attach-role-policy --role-name my-iot-role --policy-arn
 "arn:aws:iam::123456789012:policy/my-iot-policy"
```

Transmitir permissões de função

Parte de uma definição de regra é uma função do IAM que concede permissão para acessar os recursos especificados na ação da regra. O mecanismo de regras assume essa função quando a ação de regra é acionada. A função deve ser definida na mesmaConta da AWScomo regra.

Ao criar ou substituir uma regra, você está, na verdade, transferindo uma função para o mecanismo de regras. O usuário que está executando essa operação requer a permissão `iam:PassRole`. Para garantir que você tenha essa permissão, crie uma política que conceda ao`iam:PassRole`Conceder permissão e anexe-a ao usuário do IAM. A política a seguir mostra como conceder a permissão `iam:PassRole` para uma função.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Stmt1",
      "Effect": "Allow",
      "Action": [
        "iam:PassRole"
      ],
      "Resource": [
        "arn:aws:iam::123456789012:role/myRole"
```

```
        ]
      }
    ]
}
```

Nesse exemplo de política, a permissão `iam:PassRole` é concedida para a função `myRole`. A função é especificada usando o ARN da função. Você deve anexar essa política ao usuário ou função do IAM a que o usuário pertence. Para obter mais informações, consulte Trabalho com políticas gerenciadas.

> **Note**
>
> As funções Lambda usam política baseada em recursos, em que a política é anexada diretamente à própria função Lambda. Ao criar uma regra que invoca uma função do Lambda, você não transmite uma função; por isso, o usuário que está criando a regra não precisa do`iam:PassRole`permissão. Para obter mais informações sobre autorização da função Lambda, consulteConceder permissões usando uma política de recursos.

Criar uma regra do AWS IoT

Você configura regras para rotear dados das coisas conectadas. As regras são compostas de:

Nome da regra

O nome da regra.

> **Note**
>
> Não recomendamos o uso de informações de identificação do usuário em nomes de regras.

Descrição opcional

Um texto de descrição da regra.

> **Note**
>
> Não recomendamos o uso de informações de identificação do usuário em descrições de regras.

Declaração do SQL

Uma sintaxe SQL simplificada para filtrar as mensagens recebidas em um tópico do MQTT e enviar os dados para outro lugar. Para obter mais informações, consulteReferência SQL do AWS IoT (p. 568)

Versão do SQL

A versão do mecanismo de regras do SQL a ser usado ao avaliar a regra. Embora essa propriedade seja opcional, é recomendável especificar a versão do SQL. OAWS IoT CoreO console define esta propriedade como`2016-03-23`Por padrão. Se essa propriedade não for definida, como em umAWS CLIcomando ou umAWS CloudFormationModelo,`2015-10-08`é usado. Para obter mais informações, consulteVersões do SQL (p. 631)

Uma ou mais ações

As ações que a AWS IoT realiza ao executar a regra. Por exemplo, você pode inserir dados em uma tabela do DynamoDB, gravar dados em um bucket do Amazon S3, publicar em um tópico do Amazon SNS ou invocar uma função do Lambda.

Uma ação de erro

A ação que a AWS IoT executa quando não é possível executar uma ação da regra.

Ao criar uma regra, veja a quantidade de dados que você está publicando nos tópicos. Se criar regras que incluem um padrão de tópico curinga, elas poderão ser correspondidas com uma grande porcentagem de

suas mensagens, e você talvez precise aumentar a capacidade dos recursos da AWS usados pelas ações de destino. Além disso, se você criar uma regra de republicação que inclui um padrão de tópico curinga, isso poderá resultar em uma regra circular que causa um loop infinito.

> **Note**
>
> A criação e atualização de regras são ações no nível de administrador. Qualquer usuário com permissão para criar ou atualizar regras pode acessar os dados processados pelas regras.

Para criar uma regra (AWS CLI)

Use o comando create-topic-rule para criar uma regra:

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

Veja a seguir um exemplo de arquivo de carga útil com uma regra que insere todas as mensagens enviadas para `oiot/test`tópico na tabela do DynamoDB especificada. A declaração do SQL filtra as mensagens, e o ARN da função concedeAWS IoTConceder permissão para gravar na tabela do DynamoDB.

```
{
  "sql": "SELECT * FROM 'iot/test'",
  "ruleDisabled": false,
  "awsIotSqlVersion": "2016-03-23",
  "actions": [{
      "dynamoDB": {
          "tableName": "my-dynamodb-table",
          "roleArn": "arn:aws:iam::123456789012:role/my-iot-role",
          "hashKeyField": "topic",
          "hashKeyValue": "${topic(2)}",
          "rangeKeyField": "timestamp",
          "rangeKeyValue": "${timestamp()}"
      }
  }]
}
```

Veja a seguir um exemplo de arquivo de carga útil com uma regra que insere todas as mensagens enviadas para o tópico `iot/test` no bucket do S3 especificado. A declaração do SQL filtra as mensagens, e a função concede ao ARN da função concedeAWS IoTConceder permissão para gravar no bucket do Amazon S3.

```
{
  "awsIotSqlVersion": "2016-03-23",
  "sql": "SELECT * FROM 'iot/test'",
  "ruleDisabled": false,
  "actions": [
    {
      "s3": {
        "roleArn": "arn:aws:iam::123456789012:role/aws_iot_s3",
        "bucketName": "my-bucket",
        "key": "myS3Key"
      }
    }
  ]
}
```

Veja a seguir um exemplo de arquivo de carga útil com uma regra que envia dados à Amazon OpenSearch Service (Serviço da AWS):

```
{
  "sql":"SELECT *, timestamp() as timestamp FROM 'iot/test'",
```

```
  "ruleDisabled":false,
  "awsIotSqlVersion": "2016-03-23",
  "actions":[
    {
      "elasticsearch":{
        "roleArn":"arn:aws:iam::123456789012:role/aws_iot_es",
        "endpoint":"https://my-endpoint",
        "index":"my-index",
        "type":"my-type",
        "id":"${newuuid()}"
      }
    }
  ]
}
```

Veja a seguir um exemplo de arquivo de carga útil com uma regra que invoca uma função do Lambda:

```
{
  "sql": "expression",
  "ruleDisabled": false,
  "awsIotSqlVersion": "2016-03-23",
  "actions": [{
      "lambda": {
          "functionArn": "arn:aws:lambda:us-west-2:123456789012:function:my-lambda-
function"
      }
  }]
}
```

Veja a seguir um exemplo de arquivo de carga útil com uma regra que publica em um tópico do Amazon SNS:

```
{
  "sql": "expression",
  "ruleDisabled": false,
  "awsIotSqlVersion": "2016-03-23",
  "actions": [{
      "sns": {
          "targetArn": "arn:aws:sns:us-west-2:123456789012:my-sns-topic",
          "roleArn": "arn:aws:iam::123456789012:role/my-iot-role"
      }
  }]
}
```

Veja a seguir um exemplo de arquivo de carga útil com uma regra que publica novamente em outro tópico do MQTT:

```
{
  "sql": "expression",
  "ruleDisabled": false,
  "awsIotSqlVersion": "2016-03-23",
  "actions": [{
      "republish": {
          "topic": "my-mqtt-topic",
          "roleArn": "arn:aws:iam::123456789012:role/my-iot-role"
      }
  }]
}
```

Veja a seguir um exemplo de arquivo de carga útil com uma regra que envia dados para um stream do Amazon Kinesis Data Firehose:

```
{
  "sql": "SELECT * FROM 'my-topic'",
  "ruleDisabled": false,
  "awsIotSqlVersion": "2016-03-23",
  "actions": [{
      "firehose": {
          "roleArn": ""arn:aws:iam::123456789012:role/my-iot-role",
          "deliveryStreamName": "my-stream-name"
      }
  }]
}
```

Veja a seguir um exemplo de arquivo de carga útil com uma regra que usa o Amazon Machine Learning`machinelearning_predict`Se os dados na carga útil do MQTT estiverem classificados como 1.

```
{
  "sql": "SELECT * FROM 'iot/test' where machinelearning_predict('my-model',
 'arn:aws:iam::123456789012:role/my-iot-aml-role', *).predictedLabel=1",
  "ruleDisabled": false,
  "awsIotSqlVersion": "2016-03-23",
  "actions": [{
    "republish": {
        "roleArn": "arn:aws:iam::123456789012:role/my-iot-role",
        "topic": "my-mqtt-topic"
    }
  }]
}
```

O seguinte é um exemplo de arquivo de carga com uma regra que publica mensagens em um fluxo de entrada da Salesforce IoT Cloud.

```
{
  "sql": "expression",
  "ruleDisabled": false,
  "awsIotSqlVersion": "2016-03-23",
  "actions": [{
      "salesforce": {
          "token": "ABCDEFGHI123456789abcdefghi123456789",
          "url": "https://ingestion-cluster-id.my-env.sfdcnow.com/streams/stream-id/
connection-id/my-event"
      }
  }]
}
```

Veja a seguir um exemplo de arquivo de carga útil com uma regra que inicie a execução de uma máquina de estado Step Functions.

```
{
  "sql": "expression",
  "ruleDisabled": false,
  "awsIotSqlVersion": "2016-03-23",
  "actions": [{
      "stepFunctions": {
          "stateMachineName": "myCoolStateMachine",
          "executionNamePrefix": "coolRunning",
          "roleArn": "arn:aws:iam::123456789012:role/my-iot-role"
      }
  }]
}
```

Visualizar as regras

Use o comando list-topic-rules para ver as regras:

```
aws iot list-topic-rules
```

Use o comando get-topic-rule para obter informações sobre uma regra:

```
aws iot get-topic-rule --rule-name myrule
```

Excluir uma regra

Quando não precisar mais de uma regra, você poderá excluí-la.

Para excluir uma regra (AWS CLI)

Use o comando delete-topic-rule para excluir uma regra:

```
aws iot delete-topic-rule --rule-name myrule
```

Ações de regra do AWS IoT

AWS IoTAs ações da regra especificam o que fazer quando uma regra é acionada. Você pode definir ações para enviar dados para um banco de dados do Amazon DynamoDB, enviar dados para o Amazon Kinesis Data Streams, invocar umAWS Lambdafunção, e assim por diante.AWS IoTO oferece suporte às seguintes ações emRegião da AWSs onde o serviço da ação está disponível.

Ação da regra	Descrição	Nome na API
Apache Kafka (p. 487)	Envia uma mensagem para um cluster do Apache Kafka.	`kafka`
Alarmes do CloudWatch (p. 495)	Altera o estado de uma Amazon CloudWatch Alarme do	`cloudwatchAlarm`
CloudWatch Logs (p. 496)	Envia uma mensagem para a Amazon CloudWatch Logs.	`cloudwatchLogs`
Métricas do CloudWatch (p. 497)	Envie uma mensagem para um CloudWatch Métrica do.	`cloudwatchMetric`
DynamoDB (p. 499)	Envia uma mensagem para uma tabela do DynamoDB.	`dynamoDB`
DynamoDBv2 (p. 501)	Envia os dados da mensagem para várias colunas em uma tabela do DynamoDB.	`dynamoDBv2`
Elasticsearch (p. 503)	Envia uma mensagem para um endpoint do Elasticsearch.	`elasticsearch`
HTTP (p. 504)	Publicar uma mensagem em um endpoint HTTPS.	`http`

Ação da regra	Descrição	Nome na API
IoT Analytics (p. 531)	Envie uma mensagem para umAWS IoT AnalyticsChannel	`iotAnalytics`
IoT Events (p. 533)	Envie uma mensagem para umAWS IoT Eventsentrada.	`iotEvents`
IoT SiteWise (p. 534)	Envia dados de mensagens paraAWS IoT SiteWisePropriedades do ativo.	`iotSiteWise`
Kinesis Data Firehose (p. 538)	Envia uma mensagem para um fluxo de entrega do Kinesis Data Firehose.	`firehose`
Kinesis Data Streams (p. 540)	Envia uma mensagem para um fluxo de dados do Kinesis.	`kinesis`
Lambda (p. 542)	Invoca uma função do Lambda com dados de mensagem como entrada.	`lambda`
OpenSearch (p. 544)	Envia uma mensagem para uma Amazon OpenSearch Endpoint de serviço.	`OpenSearch`
Republish (p. 546)	Republica uma mensagem em outro tópico do MQTT.	`republish`
S3 (p. 547)	Armazena uma mensagem em um bucket do Amazon Simple Storage Service (Amazon S3).	`s3`
Salesforce IoT (p. 549)	Enviar uma mensagem para o fluxo de entrada da Salesforce IoT.	`salesforce`
SNS (p. 549)	Publicar uma mensagem como uma notificação por push do Amazon Simple Notification Service (Amazon SNS).	`sns`
SQS (p. 551)	Envia uma mensagem para uma fila do Amazon Simple Queue Service (Amazon SQS).	`sqs`
Step Functions (p. 553)	Inicia umAWS Step Functionsmáquina de estado do.	`stepFunctions`
the section called “Timestream” (p. 554)	Envia uma mensagem para uma tabela de banco de dados do Amazon Timestream.	`timestream`

Observações

- Você deve definir a regra na mesmaRegião da AWScomo recurso de outro serviço, para que a ação da regra possa interagir com esse recurso.

- O mecanismo de regras da AWS IoT pode fazer várias tentativas para executar uma ação em caso de erros intermitentes. Se ocorrer uma falha em todas as tentativas, a mensagem será descartada, e o erro estará disponível no CloudWatch logs. Você pode especificar uma ação de erro para cada regra que é invocada depois de ocorrer uma falha. Para obter mais informações, consulteTratamento de erros (ação de erro) (p. 565)
- Algumas ações de regra acionam ações em serviços que se integram ao AWS Key Management Service (AWS KMS) para oferecer suporte à criptografia de dados em repouso. Se você usar um gerenciado pelo clienteAWS KMS key(chave KMS) para criptografar dados em repouso, o serviço deve ter permissão para usar a chave KMS em nome do chamador. Consulte os tópicos de criptografia de dados no guia de serviço apropriado para saber como gerenciar permissões para sua chave do KMS gerenciada pelo cliente. Para obter mais informações sobre chaves KMS gerenciadas pelo cliente, consulte.AWS Key Management Serviceconceitosno AWS Key Management ServiceGuia do desenvolvedor.

Apache Kafka

A ação Apache Kafka (Kafka) envia mensagens diretamente ao seu Amazon Managed Streaming for Apache Kafka (Amazon MSK) ou clusters Apache Kafka autogerenciados para análise e visualização de dados.

> **Note**
>
> Este tópico pressupõe familiaridade com a plataforma Apache Kafka e conceitos relacionados. Para obter mais informações sobre o Apache Kafka, consulteApache Kafka.

Requisitos

Essa ação de regra tem estes requisitos:

- Uma função do IAMAWS IoTpode presumir para executar `ec2:CreateNetworkInterface`,`ec2:DescribeNetworkInterfaces`,`ec2:CreateNetworkInterfacePermi`
  `ec2:DescribeSecurityGroups`operações. Essa função cria e gerencia interfaces de rede elásticas para sua Amazon Virtual Private Cloud para alcançar seu corretor Kafka. Para obter mais informações, consulteConcessão de umAWS IoTgoverne o acesso necessário (p. 479)

  NoAWS IoTconsole, você pode escolher ou criar uma função para permitirAWS IoT CorePara executar essa ação da regra do.

  Para obter mais informações sobre interfaces de rede, consulte.Interfaces de rede elásticano Guia do usuário do Amazon EC2.

  A política anexada à função especificada deve ser semelhante ao exemplo a seguir.

```
{
    "Version": "2012-10-17",
    "Statement": [
    {
        "Effect": "Allow",
        "Action": [
            "ec2:CreateNetworkInterface",
            "ec2:DescribeNetworkInterfaces",
            "ec2:CreateNetworkInterfacePermission",
            "ec2:DeleteNetworkInterface",
            "ec2:DescribeSubnets",
            "ec2:DescribeVpcs",
            "ec2:DescribeVpcAttribute",
            "ec2:DescribeSecurityGroups"
```

```
            ],
            "Resource": "*"
        }
    ]
}
```

- Se você usarAWS Secrets ManagerPara armazenar as credenciais necessárias para se conectar à sua corretora Kafka, você deve criar uma função do IAM queAWS IoT Corepode presumir para executar `osecretsmanager:GetSecretValueesecretsmanager:DescribeSecret`operações.

  A política anexada à função especificada deve ser semelhante ao exemplo a seguir.

```
{
    "Version": "2012-10-17",
    "Statement": [
    {
        "Effect": "Allow",
        "Action": [
            "secretsmanager:GetSecretValue",
            "secretsmanager:DescribeSecret"
            ],
            "Resource": [

 "arn:aws:secretsmanager:region:123456789012:secret:kafka_client_truststore-*",
                "arn:aws:secretsmanager:region:123456789012:secret:kafka_keytab-*"
            ]
        }
    ]
}
```

- Você pode executar os clusters do Apache Kafka dentro do Amazon Virtual Private Cloud. É necessário criar um destino da Virtual Private Cloud (VPC) e usar um gateway NAT em suas sub-redes para encaminhar mensagens doAWS IoTpara um cluster público Kafka. OAWS IoTo mecanismo de regras cria uma interface de rede em cada uma das sub-redes listadas no destino da VPC para rotear o tráfego diretamente para a VPC. Ao criar um destino de VPC, oAWS IoTo mecanismo de regras cria automaticamente uma ação de regra de VPC. Para obter mais informações sobre as ações de regra da VPC, consulte.Destinos da nuvem privada virtual (VPC) (p. 492).
- Se você usar um gerenciado pelo clienteAWS KMS key(chave KMS) para criptografar dados em repouso, o serviço deve ter permissão para usar a chave KMS em nome do chamador. Para obter mais informações, consulteCriptografia do Amazon MSKnoGuia do desenvolvedor do Amazon Managed Streaming for Apache Kafka.

Parâmetros

Ao criar umAWS IoTCom essa ação, você deve especificar as seguintes informações:

DestinationArn

O nome de recurso da Amazon (ARN) do destino da VPC. Para obter informações sobre como criar um destino VPC, consulte.Destinos da nuvem privada virtual (VPC) (p. 492).

tópico

O tópico Kafka para mensagens a serem enviadas para o corretor Kafka.

Você pode substituir esse campo usando um modelo de substituição. Para obter mais informações, consultethe section called “Modelos de substituição” (p. 627)

chave (opcional)

A chave de mensagem Kafka.

Você pode substituir esse campo usando um modelo de substituição. Para obter mais informações, consultethe section called “Modelos de substituição” (p. 627)

partição (opcional)

A partição de mensagens Kafka.

Você pode substituir esse campo usando um modelo de substituição. Para obter mais informações, consultethe section called “Modelos de substituição” (p. 627)

ClientProperties

Um objeto que define as propriedades do cliente produtor Apache Kafka.

acks (opcional)

O número de reconhecimentos que o produtor exige que o servidor tenha recebido antes de considerar uma solicitação concluída.

Se você especificar 0 como o valor, o produtor não aguardará nenhuma confirmação do servidor. Se o servidor não receber a mensagem, o produtor não tentará enviar a mensagem novamente.

Valores válidos: 0, 1. O valor padrão é 1.

bootstrap.servers

Uma lista de pares de host e porta (`host1:port1`,`host2:port2`, etc.) usado para estabelecer a conexão inicial com seu cluster Kafka.

compression.type (opcional)

O tipo de compactação para todos os dados gerados pelo produtor.

Valores válidos: `none`,`gzip`,`snappy`,`lz4`,`zstd`. O valor padrão é`none`.

security.protocol

O protocolo de segurança usado para anexar ao seu corretor Kafka.

Valores válidos: `SSL`,`SASL_SSL`. O valor padrão é`SSL`.

key.serializer

Especifica como transformar os objetos-chave que você fornece com o`ProducerRecord`Em bytes do

Valor válido: `StringSerializer`.

valor.serializer

Especifica como transformar objetos de valor que você fornece com o`ProducerRecord`Em bytes do

Valor válido: `ByteBufferSerializer`.

ssl.truststore

O arquivo truststore no formato base64 ou a localização do arquivo truststore emAWS Secrets Manager. Esse valor não será obrigatório se seu truststore for confiável pelas autorias de certificados da Amazon (CA).

Esse campo é compatível com modelos de substituição. Se você usar o Secrets Manager para armazenar as credenciais necessárias para se conectar ao seu corretor Kafka, poderá usar a função get_secret SQL para recuperar o valor desse campo. Para obter mais informações sobre modelos de substituição, consulte the section called “Modelos de substituição” (p. 627). Para

obter mais informações sobre a função SQL get_secret, consulte.the section called “get_secret (secretId, SecretType, chave, roleArn)” (p. 597). Se o truststore estiver na forma de um arquivo, use o`SecretBinary`parâmetro . Se o truststore estiver na forma de uma string, use o`SecretString`parâmetro .

O tamanho máximo desse valor é 65 KB.

ssl.truststore.password

A senha para o truststore. Esse valor é necessário somente se você tiver criado uma senha para o truststore.

ssl.keystore

O arquivo de armazenamento de chaves. Esse valor é obrigatório quando você especifica`SSL`como o valor para`security.protocol`.

Esse campo é compatível com modelos de substituição. Você deve usar o Secrets Manager para armazenar as credenciais necessárias para se conectar ao seu corretor Kafka. Use a função get_secret SQL para recuperar o valor desse campo. Para obter mais informações sobre modelos de substituição, consulte the section called “Modelos de substituição” (p. 627). Para obter mais informações sobre a função SQL get_secret, consulte.the section called “get_secret (secretId, SecretType, chave, roleArn)” (p. 597). Use o parâmetro `SecretBinary`.

ssl.keystore.password

A senha de armazenamento do arquivo do repositório de chaves. Esse valor será necessário se você especificar um valor para`ssl.keystore`.

O valor desse campo pode ser texto simples. Esse campo também oferece suporte a modelos de substituição. Você deve usar o Secrets Manager para armazenar as credenciais necessárias para se conectar ao seu corretor Kafka. Use a função get_secret SQL para recuperar o valor desse campo. Para obter mais informações sobre modelos de substituição, consulte the section called “Modelos de substituição” (p. 627). Para obter mais informações sobre a função SQL get_secret, consulte.the section called “get_secret (secretId, SecretType, chave, roleArn)” (p. 597). Use o parâmetro `SecretString`.

ssl.key.password

A senha da chave privada no arquivo do armazenamento de chaves.

Esse campo é compatível com modelos de substituição. Você deve usar o Secrets Manager para armazenar as credenciais necessárias para se conectar ao seu corretor Kafka. Use a função get_secret SQL para recuperar o valor desse campo. Para obter mais informações sobre modelos de substituição, consulte the section called “Modelos de substituição” (p. 627). Para obter mais informações sobre a função SQL get_secret, consulte.the section called “get_secret (secretId, SecretType, chave, roleArn)” (p. 597). Use o parâmetro `SecretString`.

sasl.mecanismo

O mecanismo de segurança usado para se conectar ao seu corretor Kafka. Esse valor é obrigatório quando você especifica`SASL_SSL`pelo`security.protocol`.

Valores válidos: `PLAIN`, `SCRAM-SHA-512`, `GSSAPI`.

> **Note**
>
> `SCRAM-SHA-512`é o único mecanismo de segurança suportado nas regiões cn-north-1, cn-northwest-1, us-gov-east-1 e us-gov-west-1.

sasl.plain.username

O nome de usuário usado para recuperar a string secreta do Secrets Manager. Esse valor é obrigatório quando você especifica`SASL_SSL`pelo`security.protocol`e`PLAIN`pelo`sasl.mechanism`.

sasl.plain.password

A senha usada para recuperar a string secreta do Secrets Manager. Esse valor é obrigatório quando você especifica`SASL_SSL`pelo`security.protocolo``PLAIN`pelo`sasl.mechanism`.

sasl.scram.username

O nome de usuário usado para recuperar a string secreta do Secrets Manager. Esse valor é obrigatório quando você especifica`SASL_SSL`pelo`security.protocolo``SCRAM-SHA-512`pelo`sasl.mechanism`.

sasl.scram.password

A senha usada para recuperar a string secreta do Secrets Manager. Esse valor é obrigatório quando você especifica`SASL_SSL`pelo`security.protocolo``SCRAM-SHA-512`pelo`sasl.mechanism`.

sasl.kerberos.keytab

O arquivo keytab para autenticação Kerberos no Secrets Manager. Esse valor é obrigatório quando você especifica`SASL_SSL`pelo`security.protocolo``GSSAPI`pelo`sasl.mechanism`.

Esse campo é compatível com modelos de substituição. Você deve usar o Secrets Manager para armazenar as credenciais necessárias para se conectar ao seu corretor Kafka. Use a função get_secret SQL para recuperar o valor desse campo. Para obter mais informações sobre modelos de substituição, consulte the section called “Modelos de substituição” (p. 627). Para obter mais informações sobre a função SQL get_secret, consulte.the section called “get_secret (secretId, SecretType, chave, roleArn)” (p. 597). Use o parâmetro `SecretBinary`.

sasl.kerberos.service.name

O nome principal Kerberos sob o qual o Apache Kafka é executado. Esse valor é obrigatório quando você especifica`SASL_SSL`pelo`security.protocolo``GSSAPI`pelo`sasl.mechanism`.

sasl.kerberos.krb5.kdc

O nome do host do centro de distribuição de chaves (KDC) ao qual seu cliente produtor Apache Kafka se conecta. Esse valor é obrigatório quando você especifica`SASL_SSL`pelo`security.protocolo``GSSAPI`pelo`sasl.mechanism`.

sasl.kerberos.krb5.realm

O reino ao qual seu cliente produtor Apache Kafka se conecta. Esse valor é obrigatório quando você especifica`SASL_SSL`pelo`security.protocolo``GSSAPI`pelo`sasl.mechanism`.

sasl.kerberos.principal

A identidade exclusiva Kerberos à qual a Kerberos pode atribuir tickets para acessar serviços com reconhecimento de Kerberos. Esse valor é obrigatório quando você especifica`SASL_SSL`pelo`security.protocolo``GSSAPI`pelo`sasl.mechanism`.

Exemplos

O exemplo JSON a seguir define uma ação do Apache Kafka em umAWS IoTregra.

```
{
"topicRulePayload": {
    "sql": "SELECT * FROM 'some/topic'",
    "ruleDisabled": false,
    "awsIotSqlVersion": "2016-03-23",
    "actions": [
    {
        "kafka": {
```

```
            "destinationArn": "arn:aws:iot:region:123456789012:ruledestination/
vpc/VPCDestinationARN",
            "topic": "TopicName",
            "clientProperties": {
                "bootstrap.servers": "kafka.com:9092",
                "security.protocol": "SASL_SSL",
                "ssl.truststore": "${get_secret('kafka_client_truststore', 'SecretBinary',
'arn:aws:iam::123456789012:role/kafka-get-secret-role-name')}",
                "ssl.truststore.password": "kafka password",
               "sasl.mechanism": "GSSAPI",
               "sasl.kerberos.service.name": "kafka",
               "sasl.kerberos.krb5.kdc": "kerberosdns.com",
               "sasl.kerberos.keytab": "${get_secret('kafka_keytab',
               'SecretBinary', 'arn:aws:iam::123456789012:role/kafka-get-secret-role-
name')}",
               "sasl.kerberos.krb5.realm": "KERBEROSREALM",
               "sasl.kerberos.principal": "kafka-keytab/kafka-keytab.com"
            }
        }
    }
]

}
```

Notas importantes sobre a configuração do Kerberos

- Seu centro de distribuição de chaves (KDC) deve ser resolvido por meio do Sistema de Nomes de Domínio (DNS) privado dentro da VPC de destino. Uma abordagem possível é adicionar a entrada DNS do KDC a uma zona hospedada privada. Para obter mais informações sobre essa abordagem, consulteTrabalhar com zonas hospedadas privadas.
- Cada VPC deve ter a resolução DNS habilitada. Para obter mais informações, consulte Uso do DNS com sua VPC.
- Grupos de segurança de interface de rede e security groups em nível de instância no destino da VPC devem permitir o tráfego de dentro da VPC nas seguintes portas.
  - Tráfego TCP na porta do listener do corretor de inicialização (geralmente 9092, mas deve estar dentro do intervalo 9000 - 9100)
  - Tráfego TCP e UDP na porta 88 para o KDC
- `SCRAM-SHA-512`é o único mecanismo de segurança suportado nas regiões cn-north-1, cn-northwest-1, us-gov-east-1 e us-gov-west-1.

Destinos da nuvem privada virtual (VPC)

A ação de regra do Apache Kafka roteia dados para um cluster do Apache Kafka em uma Amazon Virtual Private Cloud (Amazon VPC). A configuração da VPC usada pela ação de regra do Apache Kafka é ativada automaticamente quando você especifica o destino da VPC para sua ação de regra.

Um destino da VPC contém uma lista de sub-redes dentro da VPC. O mecanismo de regras cria uma elastic network interface em cada sub-rede que você especificar nessa lista. Para obter mais informações sobre interfaces de rede, consulte.Interfaces de rede elásticaNo Guia do usuário do Amazon EC2.

Requisitos e considerações

- Se você estiver usando um cluster Apache Kafka autogerenciado que será acessado usando um endpoint público na Internet:
  - Você deve criar um gateway NAT para instâncias em suas sub-redes. O gateway NAT tem um endereço IP público que pode se conectar à Internet, o que permite que o mecanismo de regras encaminhe suas mensagens para o cluster Kafka público.

- Como uma alternativa menos dispendiosa aos gateways NAT, você pode alocar um endereço IP elástico com as interfaces de rede elásticas (ENIs) criadas pelo destino da VPC. Os security groups que você usa devem ser configurados para bloquear o tráfego de entrada.

  Note

  Se o destino da VPC estiver desativado e reativado, você deverá associar novamente os IPs elásticos aos novos ENIs.
- Se um destino de regra de tópico da VPC não receber nenhum tráfego por 30 dias seguidos, ele será desativado.
- Se algum recurso usado pelo destino da VPC mudar, o destino será desativado e não poderá ser usado.
- Algumas alterações que podem desabilitar um destino da VPC incluem: excluir a VPC, sub-redes, security groups ou a função usada; modificar a função para não ter mais as permissões necessárias e desabilitar o destino.

Preços

Para fins de definição de preço, uma ação de regra da VPC é medida além da ação que envia uma mensagem para um recurso quando o recurso está na VPC. Para obter informações sobre preços, consulte Preços do AWS IoT Core.

Criar destinos de regras de tópicos da nuvem privada virtual (VPC)

Você cria um destino da nuvem privada virtual (VPC) usando oCreateTopicRuleDestinationAPI ou o comandoAWS IoT Coreconsole do .

Ao criar um destino VPC, é necessário especificar as seguintes informações.

vpcId

O ID exclusivo do destino da VPC.

SubnetIds

Uma lista de sub-redes nas quais o mecanismo de regras cria interfaces de rede elásticas. O mecanismo de regras aloca uma única interface de rede para cada sub-rede na lista.

SecurityGroups (opcional)

Uma lista de grupos de segurança a serem aplicados às interfaces de rede.

roleArn

O nome de recurso da Amazon (ARN) de uma função que tem permissão para criar interfaces de rede em seu nome.

Esse ARN deve ter uma política anexada a ele que se pareça com o exemplo a seguir.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateNetworkInterface",
                "ec2:DescribeNetworkInterfaces",
                "ec2:DescribeVpcs",
                "ec2:DeleteNetworkInterface",
                "ec2:DescribeSubnets",
                "ec2:DescribeVpcAttribute",
```

```
                "ec2:DescribeSecurityGroups"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "ec2:CreateNetworkInterfacePermission",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "ec2:ResourceTag/VPCDestinationENI": "true"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateTags"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "ec2:CreateAction": "CreateNetworkInterface",
                    "aws:RequestTag/VPCDestinationENI": "true"
                }
            }
        }
    ]
}
```

Criando um destino VPC usandoAWS CLI

O exemplo a seguir mostra como criar um destino VPC usandoAWS CLI.

```
aws --region regions iot create-topic-rule-destination --destination-configuration
 'vpcConfiguration={subnetIds=["subnet-123456789101230456"],securityGroups=[],vpcId="vpc-
123456789101230456",roleArn="arn:aws:iam::123456789012:role/role-name"}'
```

Depois de executar esse comando, o status do destino da VPC será`IN_PROGRESS`. Após alguns minutos, seu status mudará para qualquer`ERROR`(se o comando não for bem-sucedido) ou`ENABLED`. Quando o status de destino for`ENABLED`, está pronto para uso.

Você pode usar o comando a seguir para obter o status do destino da VPC.

```
aws --region region iot get-topic-rule-destination --arn "VPCDestinationARN"
```

Criando um destino de VPC usando oAWS IoT Coreconsole

As etapas a seguir descrevem como criar um destino VPC usando a.AWS IoT Coreconsole do .

1. Navegue até o console do AWS IoT Core. No painel esquerdo, noAjaGuia, selecioneDestinos.
2. Insira valores para os seguintes campos.
   - ID da VPC
   - IDs de sub-rede

- Security group

3. Selecione uma função com as permissões necessárias para criar interfaces de rede. A política de exemplo anterior contém essas permissões.

Quando o status de destino da VPC forPERMITIDO, está pronto para uso.

Alarmes do CloudWatch

O CloudWatch Alarme (`cloudWatchAlarm`) a ação altera o estado de uma Amazon CloudWatch Alarme do Você pode especificar o motivo da alteração do estado e o valor nessa chamada.

Requisitos

Essa ação de regra tem os seguintes requisitos:

- Uma função do IAM queAWS IoTpode presumir para executar `ocloudwatch:SetAlarmState`operação. Para obter mais informações, consulteConcessão de umAWS IoTgoverne o acesso necessário (p. 479)

  NoAWS IoTconsole, você pode escolher ou criar uma função para permitirAWS IoTPara executar essa ação da regra do.

Parâmetros

Ao criar umAWS IoTCom essa ação, você deve especificar as seguintes informações:

`alarmName`

O CloudWatch Nome do alarme do

SuporteModelos de substituição (p. 627): API eAWS CLIapenas

`stateReason`

O motivo para a alteração do alarme.

SuporteModelos de substituição (p. 627): Sim

`stateValue`

O valor do estado do alarme. Valores válidos: `OK`, `ALARM`, `INSUFFICIENT_DATA`.

SuporteModelos de substituição (p. 627): Sim

`roleArn`

A função do IAM que permite o acesso ao CloudWatch Alarme do Para obter mais informações, consulteRequisitos (p. 495)

SuporteModelos de substituição (p. 627): Não

Exemplos

O exemplo JSON a seguir define um CloudWatch Ação de alarme doAWS IoTregra.

```
{
```

```
    "topicRulePayload": {
        "sql": "SELECT * FROM 'some/topic'",
        "ruleDisabled": false,
        "awsIotSqlVersion": "2016-03-23",
        "actions": [
            {
                "cloudwatchAlarm": {
                    "alarmName": "IotAlarm",
                    "stateReason": "Temperature stabilized.",
                    "stateValue": "OK",
                    "roleArn": "arn:aws:iam::123456789012:role/aws_iot_cw"
                }
            }
        ]
    }
}
```

Consulte também

- O que é o Amazon CloudWatch?noAmazônia CloudWatch Guia do usuário do
- Usar alarmes do Amazon CloudWatchnoAmazônia CloudWatch Guia do usuário do

CloudWatch Logs

O CloudWatch Logs (`cloudwatchLogs`) A ação envia dados para a Amazon CloudWatch Logs. Você pode especificar o grupo de logs do para o qual a ação envia dados.

Requisitos

Essa ação de regra tem os seguintes requisitos:

- Uma função do IAM queAWS IoTpode presumir para executar `ologs:CreateLogStream`,`logs:DescribeLogStreams`, `elogs:PutLogEvents`operações. Para obter mais informações, consulteConcessão de umAWS IoTgoverne o acesso necessário (p. 479)

  NoAWS IoTconsole, você pode escolher ou criar uma função para permitirAWS IoTPara executar essa ação da regra do.
- Se você usar um gerenciado pelo clienteAWS KMS key(chave KMS) para criptografar dados de log em CloudWatch Registros, o serviço deve ter permissão para usar a chave do KMS em nome do chamador. Para obter mais informações, consulteCriptografar dados de logs no CloudWatch Regiões usandoAWS KMSnoAmazônia CloudWatch Guia do usuário do Logs.

Parâmetros

Ao criar umAWS IoTCom essa ação, você deve especificar as seguintes informações:

`logGroupName`

O CloudWatch Grupo de logs para o qual a ação envia dados.

SuporteModelos de substituição (p. 627): API eAWS CLIapenas

`roleArn`

A função do IAM que permite o acesso ao CloudWatch grupo de logs. Para obter mais informações, consulteRequisitos (p. 496)

SuporteModelos de substituição (p. 627): Não

Exemplos

O exemplo JSON a seguir define um CloudWatch Ação de registros em umAWS IoTregra.

```
{
    "topicRulePayload": {
        "sql": "SELECT * FROM 'some/topic'",
        "ruleDisabled": false,
        "awsIotSqlVersion": "2016-03-23",
        "actions": [
            {
                "cloudwatchLogs": {
                    "logGroupName": "IotLogs",
                    "roleArn": "arn:aws:iam::123456789012:role/aws_iot_cw"
                }
            }
        ]
    }
}
```

Consulte também

- O que é o Amazon CloudWatch Logs?noAmazônia CloudWatch Guia do usuário do Logs

Métricas do CloudWatch

O CloudWatch Métrica (`cloudwatchMetric`A ação captura uma métrica do Amazon CloudWatch. Você pode especificar o namespace, nome, valor, unidade e marcação de data e hora da métrica.

Requisitos

Essa ação de regra tem os seguintes requisitos:

- Uma função do IAM queAWS IoTpode presumir para executar `ocloudwatch:PutMetricData`operação. Para obter mais informações, consulteConcessão de umAWS IoTgoverne o acesso necessário (p. 479)

  NoAWS IoTconsole, você pode escolher ou criar uma função para permitirAWS IoTPara executar essa ação da regra do.

Parâmetros

Ao criar umAWS IoTCom essa ação, você deve especificar as seguintes informações:

`metricName`

O CloudWatch Nome da métrica do.

SuporteModelos de substituição (p. 627): Sim

`metricNamespace`

O CloudWatch O nome do namespace da métrica do .

SuporteModelos de substituição (p. 627): Sim

`metricUnit`

A unidade métrica compatível com o CloudWatch.

SuporteModelos de substituição (p. 627): Sim

`metricValue`

Uma string que contém o CloudWatch valor da métrica.

SuporteModelos de substituição (p. 627): Sim

`metricTimestamp`

(Opcional) Uma string que contém o carimbo de data/hora, expressa em segundos no horário Unix epoch. O padrão é o tempo atual da época Unix.

SuporteModelos de substituição (p. 627): Sim

`roleArn`

A função do IAM que permite o acesso ao CloudWatch Métrica do. Para obter mais informações, consulteRequisitos (p. 497)

SuporteModelos de substituição (p. 627): Não

Exemplos

O exemplo JSON a seguir define um CloudWatch Ação de métrica em umAWS IoTregra.

```
{
    "topicRulePayload": {
        "sql": "SELECT * FROM 'some/topic'",
        "ruleDisabled": false,
        "awsIotSqlVersion": "2016-03-23",
        "actions": [
            {
                "cloudwatchMetric": {
                    "metricName": "IotMetric",
                    "metricNamespace": "IotNamespace",
                    "metricUnit": "Count",
                    "metricValue": "1",
                    "metricTimestamp": "1456821314",
                    "roleArn": "arn:aws:iam::123456789012:role/aws_iot_cw"
                }
            }
        ]
    }
}
```

O exemplo JSON a seguir define um CloudWatch ação métrica com modelos de substituição em umAWS IoTregra.

```
{
    "topicRulePayload": {
        "sql": "SELECT * FROM 'some/topic'",
        "ruleDisabled": false,
        "awsIotSqlVersion": "2016-03-23",
        "actions": [
            {
                "cloudwatchMetric": {
```

```
                    "metricName": "${topic()}",
                    "metricNamespace": "${namespace}",
                    "metricUnit": "${unit}",
                    "metricValue": "${value}",
                    "roleArn": "arn:aws:iam::123456789012:role/aws_iot_cw"
                }
            }
        ]
    }
}
```

Consulte também

- O que é o Amazon CloudWatch?noAmazônia CloudWatch Guia do usuário do
- Como usar métricas do Amazon CloudWatchnoAmazônia CloudWatch Guia do usuário do

DynamoDB

O DynamoDB (`dynamoDB`) grava uma mensagem MQTT inteira (ou parte dela) em uma tabela do Amazon DynamoDB.

É possível seguir um tutorial que mostra como criar e testar uma regra com uma ação do DynamoDB. Para obter mais informações, consulteTutorial: Armazenando dados do dispositivo em uma tabela do DynamoDB (p. 209)

> **Note**
>
> Essa regra grava dados que não estão no formato JSON no DynamoDB como dados binários. O console do DynamoDB exibe os dados como texto codificado em base64.

Requisitos

Essa ação de regra tem os seguintes requisitos:

- Uma função do IAM queAWS IoTpode presumir para executar `odynamodb:PutItem`operação. Para obter mais informações, consulteConcessão de umAWS IoTgoverne o acesso necessário (p. 479)

  NoAWS IoTconsole, você pode escolher ou criar uma função para permitirAWS IoTPara executar essa ação da regra do.
- Se você usar um gerenciado pelo clienteAWS KMS key(chave KMS) Para criptografar dados em repouso no DynamoDB, o serviço deve ter permissão para usar a chave KMS em nome do chamador. Para obter mais informações, consulteChave do KMS gerenciadanoGuia de conceitos básicos do Amazon DynamoDB:.

Parâmetros

Ao criar umAWS IoTCom essa ação, você deve especificar as seguintes informações:

`tableName`

O nome da tabela do DynamoDB.

SuporteModelos de substituição (p. 627): API eAWS CLIapenas

`hashKeyField`

O nome da chave de hash (também chamada de chave de partição).

SuporteModelos de substituição (p. 627): API eAWS CLIapenas

`hashKeyType`

(Opcional) O tipo de dados da chave de hash (também chamada de chave de partição). Valores válidos: `STRING`, `NUMBER`.

SuporteModelos de substituição (p. 627): API eAWS CLIapenas

`hashKeyValue`

O valor da chave de hash. Considere usar um modelo de substituição, como`${topic()}`ou`${timestamp()}`.

SuporteModelos de substituição (p. 627): Sim

`rangeKeyField`

(Opcional) O nome da chave de intervalo (também chamada de chave de classificação).

SuporteModelos de substituição (p. 627): API eAWS CLIapenas

`rangeKeyType`

(Opcional) O tipo de dados da chave de intervalo (também chamada de chave de classificação). Valores válidos: `STRING`, `NUMBER`.

SuporteModelos de substituição (p. 627): API eAWS CLIapenas

`rangeKeyValue`

(Opcional) O valor da chave de intervalo. Considere usar um modelo de substituição, como`${topic()}`ou`${timestamp()}`.

SuporteModelos de substituição (p. 627): Sim

`payloadField`

(Opcional) O nome da coluna em que a carga é gravada. Se você omitir esse valor, a carga útil será gravada na coluna chamada`payload`.

SuporteModelos de substituição (p. 627): Sim

`operation`

(Opcional) O tipo de operação a ser realizada. Valores válidos: `INSERT`, `UPDATE`, `DELETE`.

SuporteModelos de substituição (p. 627): Sim

`roleARN`

A função do IAM que permite o acesso à tabela do DynamoDB. Para obter mais informações, consulteRequisitos (p. 499)

SuporteModelos de substituição (p. 627): Não

Os dados gravados na tabela do DynamoDB são o resultado da declaração do SQL da regra.

Exemplos

O exemplo JSON a seguir define uma ação do DynamoDB em umAWS IoTregra.

```
{
    "topicRulePayload": {
```

```
        "sql": "SELECT * AS message FROM 'some/topic'",
        "ruleDisabled": false,
        "awsIotSqlVersion": "2016-03-23",
        "actions": [
            {
                "dynamoDB": {
                    "tableName": "my_ddb_table",
                    "hashKeyField": "key",
                    "hashKeyValue": "${topic()}",
                    "rangeKeyField": "timestamp",
                    "rangeKeyValue": "${timestamp()}",
                    "roleArn": "arn:aws:iam::123456789012:role/aws_iot_dynamoDB"
                }
            }
        ]
    }
}
```

Consulte também

- O que é o Amazon DynamoDB?noGuia do desenvolvedor do Amazon DynamoDB
- Conceitos básicos do DynamoDBnoGuia do desenvolvedor do Amazon DynamoDB
- Tutorial: Armazenando dados do dispositivo em uma tabela do DynamoDB (p. 209)

DynamoDBv2

O DynamoDBv2`dynamoDBv2`) grava uma mensagem MQTT inteira (ou parte dela) em uma tabela do Amazon DynamoDB. Cada atributo na carga útil é gravado em uma coluna separada no banco de dados DynamoDB.

Requisitos

Essa ação de regra tem os seguintes requisitos:

- Uma função do IAM queAWS IoTpode presumir para executar `odynamodb:PutItem`operação. Para obter mais informações, consulteConcessão de umAWS IoTgoverne o acesso necessário (p. 479)

  NoAWS IoTconsole, você pode escolher ou criar uma função para permitirAWS IoTPara executar essa ação da regra do.
- A carga útil da mensagem MQTT deve conter uma chave de nível raiz que corresponda à chave de partição primária da tabela e uma chave de nível raiz que corresponda à chave de classificação primária da tabela, caso esteja definida.
- Se você usar um gerenciado pelo clienteAWS KMS key(chave KMS) Para criptografar dados em repouso no DynamoDB, o serviço deve ter permissão para usar a chave KMS em nome do chamador. Para obter mais informações, consulteChave do KMS gerenciadanoGuia de conceitos básicos do Amazon DynamoDB:.

Parâmetros

Ao criar umAWS IoTCom essa ação, você deve especificar as seguintes informações:

`putItem`

Um objeto que especifica a tabela do DynamoDB na qual os dados da mensagem serão gravados. Esse objeto deve conter as seguintes informações:

`tableName`

O nome da tabela do DynamoDB.

SuporteModelos de substituição (p. 627): API eAWS CLIapenas

`roleARN`

A função do IAM que permite o acesso à tabela do DynamoDB. Para obter mais informações, consulteRequisitos (p. 501)

SuporteModelos de substituição (p. 627): Não

Os dados gravados na tabela do DynamoDB são o resultado da declaração do SQL da regra.

Exemplos

O exemplo JSON a seguir define uma ação do DynamoDBV2 em umAWS IoTregra.

```
{
    "topicRulePayload": {
        "sql": "SELECT * AS message FROM 'some/topic'",
        "ruleDisabled": false,
        "awsIotSqlVersion": "2016-03-23",
        "actions": [
            {
                "dynamoDBv2": {
                    "putItem": {
                        "tableName": "my_ddb_table"
                    },
                    "roleArn": "arn:aws:iam::123456789012:role/aws_iot_dynamoDBv2",
                }
            }
        ]
    }
}
```

O exemplo JSON a seguir define uma ação do DynamoDB com modelos de substituição em umAWS IoTregra.

```
{
    "topicRulePayload": {
        "sql": "SELECT * FROM 'some/topic'",
        "ruleDisabled": false,
        "awsIotSqlVersion": "2015-10-08",
        "actions": [
            {
                "dynamoDBv2": {
                    "putItem": {
                        "tableName": "${topic()}"
                    },
                    "roleArn": "arn:aws:iam::123456789012:role/aws_iot_dynamoDBv2"
                }
            }
        ]
    }
}
```

Consulte também

- O que é o Amazon DynamoDB?noGuia do desenvolvedor do Amazon DynamoDB

- Conceitos básicos do DynamoDBnoGuia do desenvolvedor do Amazon DynamoDB

Elasticsearch

O Elasticsearch (`elasticsearch`) grava dados de mensagens MQTT em uma Amazon OpenSearch Domínio de serviço. Você pode usar ferramentas como OpenSearch Painéis para consultar e visualizar dados em OpenSearch Serviço.

> Warning
>
> O`Elasticsearch`A ação só pode ser usada por ações de regra existentes. Para criar uma nova ação de regra ou atualizar uma ação de regra existente, use o`OpenSearch`Ação de regra do. Para obter mais informações, consulteOpenSearch (p. 544)

Requisitos

Essa ação de regra tem os seguintes requisitos:

- Uma função do IAM queAWS IoTpode presumir para executar `es:ESHttpPut`operação. Para obter mais informações, consulteConcessão de umAWS IoTgoverne o acesso necessário (p. 479)

  NoAWS IoTconsole, você pode escolher ou criar uma função para permitirAWS IoTPara executar essa ação da regra do.
- Se você usar um gerenciado pelo clienteAWS KMS key(chave KMS) Para criptografar dados em repouso no OpenSearch, o serviço deve ter permissão para usar a chave KMS em nome do chamador. Para obter mais informações, consulteCriptografia de dados em repouso para a Amazon OpenSearch ServiçonoAmazônia OpenSearch Guia do desenvolvedor de serviço.

Parâmetros

Ao criar umAWS IoTCom essa ação, você deve especificar as seguintes informações:

`endpoint`

O endpoint do domínio do serviço.

SuporteModelos de substituição (p. 627): API eAWS CLIapenas

`index`

O índice do em que você deseja armazenar seus dados.

SuporteModelos de substituição (p. 627): Sim

`type`

O tipo de documento que você está armazenando.

SuporteModelos de substituição (p. 627): Sim

`id`

O identificador exclusivo de cada documento.

SuporteModelos de substituição (p. 627): Sim

`roleARN`

A função do IAM que permite o acesso ao OpenSearch Domínio de serviço. Para obter mais informações, consulteRequisitos (p. 503)

SuporteModelos de substituição (p. 627): Não

Exemplos

O exemplo JSON a seguir define uma ação do Elasticsearch em umAWS IoTregra e como você pode especificar os campos para `oelasticsearch`Ação . Para obter mais informações, consulteElasticsearchAction.

```
{
    "topicRulePayload": {
        "sql": "SELECT *, timestamp() as timestamp FROM 'iot/test'",
        "ruleDisabled": false,
        "awsIotSqlVersion": "2016-03-23",
        "actions": [
            {
                "elasticsearch": {
                    "endpoint": "https://my-endpoint",
                    "index": "my-index",
                    "type": "my-type",
                    "id": "${newuuid()}",
                    "roleArn": "arn:aws:iam::123456789012:role/aws_iot_es"
                }
            }
        ]
    }
}
```

O exemplo JSON a seguir define uma ação do Elasticsearch com modelos de substituição em umAWS IoTregra.

```
{
    "topicRulePayload": {
        "sql": "SELECT * FROM 'some/topic'",
        "ruleDisabled": false,
        "awsIotSqlVersion": "2016-03-23",
        "actions": [
            {
                "elasticsearch": {
                    "endpoint": "https://my-endpoint",
                    "index": "${topic()}",
                    "type": "${type}",
                    "id": "${newuuid()}",
                    "roleArn": "arn:aws:iam::123456789012:role/aws_iot_es"
                }
            }
        ]
    }
}
```

Consulte também

- OpenSearch (p. 544)
- O que é o Amazon OpenSearch Serviço?

HTTP

O HTTPS (`http`) A ação envia os dados de uma mensagem MQTT para um serviço ou aplicativo web.

Requisitos

Essa ação de regra tem os seguintes requisitos:

- Você deve confirmar e habilitar endpoints HTTPS antes que o mecanismo de regras possa usá-los. Para obter mais informações, consulteTrabalhar com destinos de regra de tópico HTTP (p. 507)

Parâmetros

Ao criar umAWS IoTCom essa ação, você deve especificar as seguintes informações:

`url`

O endpoint HTTPS onde a mensagem é enviada usando o método POST HTTP. Se você usar um endereço IP no lugar de um nome de host, ele deve ser um endereço IPv4. Os endereços IPv6 não são suportados.

SuporteModelos de substituição (p. 627): Sim

`confirmationUrl`

(Opcional) Se especificado,AWS IoTUse a URL de confirmação para criar um destino de regra de tópico correspondente. Você deve habilitar o destino da regra de tópico antes de usá-lo em uma ação HTTP. Para obter mais informações, consulteTrabalhar com destinos de regra de tópico HTTP (p. 507) Se você usar modelos de substituição, deverá criar manualmente destinos de regra de tópico antes que a ação `http` possa ser usada. `confirmationUrl` deve ser um prefixo de `url`.

A relação entre `url` e `confirmationUrl` é descrita pelo seguinte:

- Se `url` for codificado e não `confirmationUrl` for fornecido, tratamos implicitamente o campo `url` como `confirmationUrl`. A AWS IoT cria um destino de regra de tópico para `url`.
- Se `url` e `confirmationUrl` forem codificados, `url` deverá começar com `confirmationUrl`. A AWS IoT cria um destino de regra de tópico para `confirmationUrl`.
- Se `url` contiver um modelo de substituição, você deverá especificar `confirmationUrl` e `url` deverá começar com `confirmationUrl`. Se `confirmationUrl` contiver modelos de substituição, você deverá criar manualmente destinos de regra de tópico antes que a ação `http` possa ser usada. Se `confirmationUrl` não contiver modelos de substituição, a AWS IoT criará um destino de regra de tópico para `confirmationUrl`.

SuporteModelos de substituição (p. 627): Sim

`headers`

(Opcional) A lista de cabeçalhos a serem incluídos em solicitações HTTP para o endpoint. Cada cabeçalho deve conter as seguintes informações:

`key`

A chave do cabeçalho.

SuporteModelos de substituição (p. 627): Não

`value`

O valor do cabeçalho.

SuporteModelos de substituição (p. 627): Sim

Note

O tipo de conteúdo padrão é application/json quando a carga está no formato JSON. Caso contrário, é application/octet-stream. Você pode substituí-lo especificando o tipo de conteúdo

exato no cabeçalho com o tipo de conteúdo da chave (sem distinção entre maiúsculas e minúsculas).

`auth`

(Opcional) A autenticação usada pelo mecanismo de regras para se conectar à URL do endpoint especificado no`url`Argumento do Atualmente, o Signature versão 4 é o único tipo de autenticação suportado. Para obter mais informações, consulte Autorização HTTP.

SuporteModelos de substituição (p. 627): Não

Exemplos

O exemplo JSON a seguir define umAWS IoTregra com uma ação HTTP.

```
{
    "topicRulePayload": {
        "sql": "SELECT * FROM 'some/topic'",
        "ruleDisabled": false,
        "awsIotSqlVersion": "2016-03-23",
        "actions": [
            {
                "http": {
                    "url": "https://www.example.com/subpath",
                    "confirmationUrl": "https://www.example.com",
                    "headers": [
                        {
                            "key": "static_header_key",
                            "value": "static_header_value"
                        },
                        {
                            "key": "substitutable_header_key",
                            "value": "${value_from_payload}"
                        }
                    ]
                }
            }
        ]
    }
}
```

Lógica de nova tentativa de ação HTTP

OAWS IoTO mecanismo de regras repete a ação HTTP de acordo com estas regras:

- O mecanismo de regras tenta enviar uma mensagem pelo menos uma vez.
- O mecanismo de regras tenta novamente no máximo duas vezes. O número máximo de tentativas é três.
- O mecanismo de regras não fará uma nova tentativa se:
  - A tentativa anterior tiver fornecido uma resposta maior do que 16384 bytes.
  - O serviço ou aplicativo web downstream fechar a conexão TCP após a tentativa.
  - O tempo total para concluir uma solicitação com repetições tiver excedido o limite de tempo limite da solicitação.
  - A solicitação retorna um código de status HTTP diferente de 429, 500–599.

> **Note**
>
> Custos padrão de transferência de dados se aplicam a novas tentativas.

Consulte também

- Trabalhar com destinos de regra de tópico HTTP (p. 507)
- Encaminhar dados diretamente doAWS IoT Corepara seus serviços da WebnoInternet das Coisas noAWSblog

Trabalhar com destinos de regra de tópico HTTP

Um destino de regra de tópico HTTP é um serviço da Web para o qual o mecanismo de regras pode rotear dados de uma regra de tópico. UmaAWS IoT CoreO recurso descreve o serviço Web doAWS IoT. Os recursos de destino da regra de tópico podem ser compartilhados por regras diferentes.

AntesAWS IoT Corepode enviar dados para outro serviço da Web, ele deve confirmar que ele pode acessar o endpoint do serviço.

Visão geral do destino da regra de tópico HTTP

Um destino de regra de tópico HTTP refere-se a um serviço da Web que oferece suporte a um URL de confirmação e um ou mais URLs de coleta de dados. O recurso de destino da regra de tópico HTTP contém o URL de confirmação do seu serviço Web. Ao configurar uma ação de regra de tópico HTTP, você especifica o URL real do endpoint que deve receber os dados junto com o URL de confirmação do serviço Web. Depois que seu destino for confirmado, a regra de tópico envia o resultado da instrução SQL para o endpoint HTTPS (e não para o URL de confirmação).

Um destino de regra de tópico HTTP pode estar em um dos seguintes estados:

ENABLED

O destino foi confirmado e pode ser usado por uma ação de regra. Um destino deve estar no estado `ENABLED` para que ser usado em uma regra. Você só pode habilitar um destino com status DISABLED.

DISABLED

O destino foi confirmado, mas não pode ser usado por uma ação de regra. Isso é útil se você quiser impedir temporariamente o tráfego para seu endpoint sem precisar passar pelo processo de confirmação novamente. Você só pode desabilitar um destino com status ENABLED.

IN_PROGRESS

A confirmação do destino está em andamento.

ERROR

A confirmação do destino expirou.

Depois que um destino de regra de tópico HTTP foi confirmado e habilitado, ele poderá ser usado com qualquer regra em sua conta.

As seções a seguir descrevem ações comuns nos destinos de regras de tópicos HTTP.

Criando e confirmando destinos de regras de tópicos HTTP

Você cria um destino de regra de tópico HTTP chamando o`CreateTopicRuleDestination`operação ou usando oAWS IoTconsole do .

Depois de criar um destino,AWS IoTenvia uma solicitação de confirmação para o URL de confirmação. A solicitação de confirmação tem o seguinte formato:

```
HTTP POST {confirmationUrl}/?confirmationToken={confirmationToken}
Headers:
x-amz-rules-engine-message-type: DestinationConfirmation
x-amz-rules-engine-destination-arn:"arn:aws:iot:us-east-1:123456789012:ruledestination/
http/7a280e37-b9c6-47a2-a751-0703693f46e4"
Content-Type: application/json
Body:
{
    "arn":"arn:aws:iot:us-east-1:123456789012:ruledestination/http/7a280e37-b9c6-47a2-
a751-0703693f46e4",
    "confirmationToken": "AYADeMXLrPrNY2wqJAKsFNn-…NBJndA",
    "enableUrl": "https://iot.us-east-1.amazonaws.com/confirmdestination/
AYADeMXLrPrNY2wqJAKsFNn-…NBJndA",
    "messageType": "DestinationConfirmation"
}
```

O conteúdo da solicitação de confirmação inclui as seguintes informações:

arn

O nome de recurso da Amazon (ARN) do destino da regra de tópico a ser confirmado.

confirmationToken

O token de confirmação enviado porAWS IoT Core. O token no exemplo é truncado. O token será mais longo. Você precisará desse token para confirmar seu destino comAWS IoT Core.

enableUrl

O URL para o qual você navega para confirmar um destino de regra de tópico.

messageType

O tipo de mensagem.

Para concluir o processo de confirmação do endpoint, você deve executar um dos seguintes procedimentos depois que o URL de confirmação receber a solicitação de confirmação.

- Chame o`enableUrl`na solicitação de confirmação e, em seguida, ligue`UpdateTopicRuleDestination`Para definir o status da regra de tópico como`ENABLED`.
- Chame o`ConfirmTopicRuleDestination`operação e passagem do`confirmationToken`a partir da solicitação de confirmação.
- Copie o`confirmationToken`e cole-o na caixa de diálogo de confirmação do destino naAWS IoTconsole do .

Envio de uma nova solicitação de confirmação

Para acionar uma nova mensagem de confirmação para um destino, chame `UpdateTopicRuleDestination` e defina o status do destino da regra de tópico como `IN_PROGRESS`.

Você precisará repetir o processo de confirmação depois de enviar uma nova solicitação de confirmação

Desativação e exclusão de um destino de regra de tópico

Para desativar um destino, chame `UpdateTopicRuleDestination` e defina o status do destino da regra de tópico como `DISABLED`. Uma regra de tópico no estado DISABLED pode ser ativada novamente sem a necessidade de enviar uma nova solicitação de confirmação.

Para excluir um destino de regra de tópico, chame `DeleteTopicRuleDestination`.

Autoridades de certificação suportadas por endpoints HTTPS em destinos de regras de tópico

As autoridades de certificação a seguir são suportadas por endpoints HTTPS em destinos de regras de tópico.

```
Alias name: swisssignplatinumg2ca
Certificate fingerprints:
  MD5:  C9:98:27:77:28:1E:3D:0E:15:3C:84:00:B8:85:03:E6
  SHA1: 56:E0:FA:C0:3B:8F:18:23:55:18:E5:D3:11:CA:E8:C2:43:31:AB:66
  SHA256:
 3B:22:2E:56:67:11:E9:92:30:0D:C0:B1:5A:B9:47:3D:AF:DE:F8:C8:4D:0C:EF:7D:33:17:B4:C1:82:1D:14:36

Alias name: hellenicacademicandresearchinstitutionsrootca2011
Certificate fingerprints:
  MD5:  73:9F:4C:4B:73:5B:79:E9:FA:BA:1C:EF:6E:CB:D5:C9
  SHA1: FE:45:65:9B:79:03:5B:98:A1:61:B5:51:2E:AC:DA:58:09:48:22:4D
  SHA256:
 BC:10:4F:15:A4:8B:E7:09:DC:A5:42:A7:E1:D4:B9:DF:6F:05:45:27:E8:02:EA:A9:2D:59:54:44:25:8A:FE:71

Alias name: teliasonerarootcav1
Certificate fingerprints:
  MD5:  37:41:49:1B:18:56:9A:26:F5:AD:C2:66:FB:40:A5:4C
  SHA1: 43:13:BB:96:F1:D5:86:9B:C1:4E:6A:92:F6:CF:F6:34:69:87:82:37
  SHA256:
 DD:69:36:FE:21:F8:F0:77:C1:23:A1:A5:21:C1:22:24:F7:22:55:B7:3E:03:A7:26:06:93:E8:A2:4B:0F:A3:89

Alias name: geotrustprimarycertificationauthority
Certificate fingerprints:
  MD5:  02:26:C3:01:5E:08:30:37:43:A9:D0:7D:CF:37:E6:BF
  SHA1: 32:3C:11:8E:1B:F7:B8:B6:52:54:E2:E2:10:0D:D6:02:90:37:F0:96
  SHA256:
 37:D5:10:06:C5:12:EA:AB:62:64:21:F1:EC:8C:92:01:3F:C5:F8:2A:E9:8E:E5:33:EB:46:19:B8:DE:B4:D0:6C

Alias name: trustisfpsrootca
Certificate fingerprints:
  MD5:  30:C9:E7:1E:6B:E6:14:EB:65:B2:16:69:20:31:67:4D
  SHA1: 3B:C0:38:0B:33:C3:F6:A6:0C:86:15:22:93:D9:DF:F5:4B:81:C0:04
  SHA256:
 C1:B4:82:99:AB:A5:20:8F:E9:63:0A:CE:55:CA:68:A0:3E:DA:5A:51:9C:88:02:A0:D3:A6:73:BE:8F:8E:55:7D

Alias name: quovadisrootca3g3
Certificate fingerprints:
  MD5:  DF:7D:B9:AD:54:6F:68:A1:DF:89:57:03:97:43:B0:D7
  SHA1: 48:12:BD:92:3C:A8:C4:39:06:E7:30:6D:27:96:E6:A4:CF:22:2E:7D
  SHA256:
 88:EF:81:DE:20:2E:B0:18:45:2E:43:F8:64:72:5C:EA:5F:BD:1F:C2:D9:D2:05:73:07:09:C5:D8:B8:69:0F:46

Alias name: buypassclass2ca
Certificate fingerprints:
  MD5:  46:A7:D2:FE:45:FB:64:5A:A8:59:90:9B:78:44:9B:29
  SHA1: 49:0A:75:74:DE:87:0A:47:FE:58:EE:F6:C7:6B:EB:C6:0B:12:40:99
  SHA256:
 9A:11:40:25:19:7C:5B:B9:5D:94:E6:3D:55:CD:43:79:08:47:B6:46:B2:3C:DF:11:AD:A4:A0:0E:FF:15:FB:48

Alias name: secureglobalca
Certificate fingerprints:
  MD5:  CF:F4:27:0D:D4:ED:DC:65:16:49:6D:3D:DA:BF:6E:DE
  SHA1: 3A:44:73:5A:E5:81:90:1F:24:86:61:46:1E:3B:9C:C4:5F:F5:3A:1B
  SHA256:
 42:00:F5:04:3A:C8:59:0E:BB:52:7D:20:9E:D1:50:30:29:FB:CB:D4:1C:A1:B5:06:EC:27:F1:5A:DE:7D:AC:69

Alias name: chunghwaepkirootca
```

```
Certificate fingerprints:
  MD5:  1B:2E:00:CA:26:06:90:3D:AD:FE:6F:15:68:D3:6B:B3
  SHA1: 67:65:0D:F1:7E:8E:7E:5B:82:40:A4:F4:56:4B:CF:E2:3D:69:C6:F0
  SHA256:
 C0:A6:F4:DC:63:A2:4B:FD:CF:54:EF:2A:6A:08:2A:0A:72:DE:35:80:3E:2F:F5:FF:52:7A:E5:D8:72:06:DF:D5

Alias name: verisignclass2g2ca
Certificate fingerprints:
  MD5:  2D:BB:E5:25:D3:D1:65:82:3A:B7:0E:FA:E6:EB:E2:E1
  SHA1: B3:EA:C4:47:76:C9:C8:1C:EA:F2:9D:95:B6:CC:A0:08:1B:67:EC:9D
  SHA256:
 3A:43:E2:20:FE:7F:3E:A9:65:3D:1E:21:74:2E:AC:2B:75:C2:0F:D8:98:03:05:BC:50:2C:AF:8C:2D:9B:41:A1

Alias name: szafirrootca2
Certificate fingerprints:
  MD5:  11:64:C1:89:B0:24:B1:8C:B1:07:7E:89:9E:51:9E:99
  SHA1: E2:52:FA:95:3F:ED:DB:24:60:BD:6E:28:F3:9C:CC:CF:5E:B3:3F:DE
  SHA256:
 A1:33:9D:33:28:1A:0B:56:E5:57:D3:D3:2B:1C:E7:F9:36:7E:B0:94:BD:5F:A7:2A:7E:50:04:C8:DE:D7:CA:FE

Alias name: quovadisrootca1g3
Certificate fingerprints:
  MD5:  A4:BC:5B:3F:FE:37:9A:FA:64:F0:E2:FA:05:3D:0B:AB
  SHA1: 1B:8E:EA:57:96:29:1A:C9:39:EA:B8:0A:81:1A:73:73:C0:93:79:67
  SHA256:
 8A:86:6F:D1:B2:76:B5:7E:57:8E:92:1C:65:82:8A:2B:ED:58:E9:F2:F2:88:05:41:34:B7:F1:F4:BF:C9:CC:74

Alias name: utndatacorpsgcca
Certificate fingerprints:
  MD5:  B3:A5:3E:77:21:6D:AC:4A:C0:C9:FB:D5:41:3D:CA:06
  SHA1: 58:11:9F:0E:12:82:87:EA:50:FD:D9:87:45:6F:4F:78:DC:FA:D6:D4
  SHA256:
 85:FB:2F:91:DD:12:27:5A:01:45:B6:36:53:4F:84:02:4A:D6:8B:69:B8:EE:88:68:4F:F7:11:37:58:05:B3:48

Alias name: autoridaddecertificacionfirmaprofesionalcifa62634068
Certificate fingerprints:
  MD5:  73:3A:74:7A:EC:BB:A3:96:A6:C2:E4:E2:C8:9B:C0:C3
  SHA1: AE:C5:FB:3F:C8:E1:BF:C4:E5:4F:03:07:5A:9A:E8:00:B7:F7:B6:FA
  SHA256:
 04:04:80:28:BF:1F:28:64:D4:8F:9A:D4:D8:32:94:36:6A:82:88:56:55:3F:3B:14:30:3F:90:14:7F:5D:40:EF

Alias name: securesignrootca11
Certificate fingerprints:
  MD5:  B7:52:74:E2:92:B4:80:93:F2:75:E4:CC:D7:F2:EA:26
  SHA1: 3B:C4:9F:48:F8:F3:73:A0:9C:1E:BD:F8:5B:B1:C3:65:C7:D8:11:B3
  SHA256:
 BF:0F:EE:FB:9E:3A:58:1A:D5:F9:E9:DB:75:89:98:57:43:D2:61:08:5C:4D:31:4F:6F:5D:72:59:AA:42:16:12

Alias name: amazon-ca-g4-acm2
Certificate fingerprints:
  MD5:  B2:F1:03:2B:93:64:05:80:B8:A8:17:36:B9:1B:52:3C
  SHA1: A7:E6:45:32:1F:7A:B7:AD:C0:70:EA:73:5F:AB:ED:C3:DA:B4:D0:C8
  SHA256:
 D7:A8:7C:69:95:D0:E2:04:2A:32:70:A7:E2:87:FE:A7:E8:F4:C1:70:62:F7:90:C3:EB:BB:53:F2:AC:39:26:BE

Alias name: isrgrootx1
Certificate fingerprints:
  MD5:  0C:D2:F9:E0:DA:17:73:E9:ED:86:4D:A5:E3:70:E7:4E
  SHA1: CA:BD:2A:79:A1:07:6A:31:F2:1D:25:36:35:CB:03:9D:43:29:A5:E8
  SHA256:
 96:BC:EC:06:26:49:76:F3:74:60:77:9A:CF:28:C5:A7:CF:E8:A3:C0:AA:E1:1A:8F:FC:EE:05:C0:BD:DF:08:C6

Alias name: amazon-ca-g4-acm1
Certificate fingerprints:
  MD5:  E2:F1:18:19:61:5C:43:E0:D4:A8:5D:0B:FA:7C:89:1B
  SHA1: F2:0D:28:B6:29:C2:2C:5E:84:05:E6:02:4D:97:FE:8F:A0:84:93:A0
```

```
  SHA256:
 B0:11:A4:F7:29:6C:74:D8:2B:F5:62:DF:87:D7:28:C7:1F:B5:8C:F4:E6:73:F2:78:FC:DA:F3:FF:83:A6:8C:87

Alias name: etugracertificationauthority
Certificate fingerprints:
  MD5:  B8:A1:03:63:B0:BD:21:71:70:8A:6F:13:3A:BB:79:49
  SHA1: 51:C6:E7:08:49:06:6E:F3:92:D4:5C:A0:0D:6D:A3:62:8F:C3:52:39
  SHA256:
 B0:BF:D5:2B:B0:D7:D9:BD:92:BF:5D:4D:C1:3D:A2:55:C0:2C:54:2F:37:83:65:EA:89:39:11:F5:5E:55:F2:3C

Alias name: geotrustuniversalca2
Certificate fingerprints:
  MD5:  34:FC:B8:D0:36:DB:9E:14:B3:C2:F2:DB:8F:E4:94:C7
  SHA1: 37:9A:19:7B:41:85:45:35:0C:A6:03:69:F3:3C:2E:AF:47:4F:20:79
  SHA256:
 A0:23:4F:3B:C8:52:7C:A5:62:8E:EC:81:AD:5D:69:89:5D:A5:68:0D:C9:1D:1C:B8:47:7F:33:F8:78:B9:5B:0B

Alias name: digicertglobalrootca
Certificate fingerprints:
  MD5:  79:E4:A9:84:0D:7D:3A:96:D7:C0:4F:E2:43:4C:89:2E
  SHA1: A8:98:5D:3A:65:E5:E5:C4:B2:D7:D6:6D:40:C6:DD:2F:B1:9C:54:36
  SHA256:
 43:48:A0:E9:44:4C:78:CB:26:5E:05:8D:5E:89:44:B4:D8:4F:96:62:BD:26:DB:25:7F:89:34:A4:43:C7:01:61

Alias name: staatdernederlandenevrootca
Certificate fingerprints:
  MD5:  FC:06:AF:7B:E8:1A:F1:9A:B4:E8:D2:70:1F:C0:F5:BA
  SHA1: 76:E2:7E:C1:4F:DB:82:C1:C0:A6:75:B5:05:BE:3D:29:B4:ED:DB:BB
  SHA256:
 4D:24:91:41:4C:FE:95:67:46:EC:4C:EF:A6:CF:6F:72:E2:8A:13:29:43:2F:9D:8A:90:7A:C4:CB:5D:AD:C1:5A

Alias name: utnuserfirstclientauthemailca
Certificate fingerprints:
  MD5:  D7:34:3D:EF:1D:27:09:28:E1:31:02:5B:13:2B:DD:F7
  SHA1: B1:72:B1:A5:6D:95:F9:1F:E5:02:87:E1:4D:37:EA:6A:44:63:76:8A
  SHA256:
 43:F2:57:41:2D:44:0D:62:74:76:97:4F:87:7D:A8:F1:FC:24:44:56:5A:36:7A:E6:0E:DD:C2:7A:41:25:31:AE

Alias name: actalisauthenticationrootca
Certificate fingerprints:
  MD5:  69:C1:0D:4F:07:A3:1B:C3:FE:56:3D:04:BC:11:F6:A6
  SHA1: F3:73:B3:87:06:5A:28:84:8A:F2:F3:4A:CE:19:2B:DD:C7:8E:9C:AC
  SHA256:
 55:92:60:84:EC:96:3A:64:B9:6E:2A:BE:01:CE:0B:A8:6A:64:FB:FE:BC:C7:AA:B5:AF:C1:55:B3:7F:D7:60:66

Alias name: amazonrootca4
Certificate fingerprints:
  MD5:  89:BC:27:D5:EB:17:8D:06:6A:69:D5:FD:89:47:B4:CD
  SHA1: F6:10:84:07:D6:F8:BB:67:98:0C:C2:E2:44:C2:EB:AE:1C:EF:63:BE
  SHA256:
 E3:5D:28:41:9E:D0:20:25:CF:A6:90:38:CD:62:39:62:45:8D:A5:C6:95:FB:DE:A3:C2:2B:0B:FB:25:89:70:92

Alias name: amazonrootca3
Certificate fingerprints:
  MD5:  A0:D4:EF:0B:F7:B5:D8:49:95:2A:EC:F5:C4:FC:81:87
  SHA1: 0D:44:DD:8C:3C:8C:1A:1A:58:75:64:81:E9:0F:2E:2A:FF:B3:D2:6E
  SHA256:
 18:CE:6C:FE:7B:F1:4E:60:B2:E3:47:B8:DF:E8:68:CB:31:D0:2E:BB:3A:DA:27:15:69:F5:03:43:B4:6D:B3:A4

Alias name: amazonrootca2
Certificate fingerprints:
  MD5:  C8:E5:8D:CE:A8:42:E2:7A:C0:2A:5C:7C:9E:26:BF:66
  SHA1: 5A:8C:EF:45:D7:A6:98:59:76:7A:8C:8B:44:96:B5:78:CF:47:4B:1A
  SHA256:
 1B:A5:B2:AA:8C:65:40:1A:82:96:01:18:F8:0B:EC:4F:62:30:4D:83:CE:C4:71:3A:19:C3:9C:01:1E:A4:6D:B4
```

```
Alias name: amazonrootca1
Certificate fingerprints:
  MD5:  43:C6:BF:AE:EC:FE:AD:2F:18:C6:88:68:30:FC:C8:E6
  SHA1: 8D:A7:F9:65:EC:5E:FC:37:91:0F:1C:6E:59:FD:C1:CC:6A:6E:DE:16
  SHA256:
 8E:CD:E6:88:4F:3D:87:B1:12:5B:A3:1A:C3:FC:B1:3D:70:16:DE:7F:57:CC:90:4F:E1:CB:97:C6:AE:98:19:6E

Alias name: affirmtrustpremium
Certificate fingerprints:
  MD5:  C4:5D:0E:48:B6:AC:28:30:4E:0A:BC:F9:38:16:87:57
  SHA1: D8:A6:33:2C:E0:03:6F:B1:85:F6:63:4F:7D:6A:06:65:26:32:28:27
  SHA256:
 70:A7:3F:7F:37:6B:60:07:42:48:90:45:34:B1:14:82:D5:BF:0E:69:8E:CC:49:8D:F5:25:77:EB:F2:E9:3B:9A

Alias name: keynectisrootca
Certificate fingerprints:
  MD5:  CC:4D:AE:FB:30:6B:D8:38:FE:50:EB:86:61:4B:D2:26
  SHA1: 9C:61:5C:4D:4D:85:10:3A:53:26:C2:4D:BA:EA:E4:A2:D2:D5:CC:97
  SHA256:
 42:10:F1:99:49:9A:9A:C3:3C:8D:E0:2B:A6:DB:AA:14:40:8B:DD:8A:6E:32:46:89:C1:92:2D:06:97:15:A3:32

Alias name: equifaxsecureglobalebusinessca1
Certificate fingerprints:
  MD5:  51:F0:2A:33:F1:F5:55:39:07:F2:16:7A:47:C7:5D:63
  SHA1: 3A:74:CB:7A:47:DB:70:DE:89:1F:24:35:98:64:B8:2D:82:BD:1A:36
  SHA256:
 86:AB:5A:65:71:D3:32:9A:BC:D2:E4:E6:37:66:8B:A8:9C:73:1E:C2:93:B6:CB:A6:0F:71:63:40:A0:91:CE:AE

Alias name: affirmtrustpremiumca
Certificate fingerprints:
  MD5:  C4:5D:0E:48:B6:AC:28:30:4E:0A:BC:F9:38:16:87:57
  SHA1: D8:A6:33:2C:E0:03:6F:B1:85:F6:63:4F:7D:6A:06:65:26:32:28:27
  SHA256:
 70:A7:3F:7F:37:6B:60:07:42:48:90:45:34:B1:14:82:D5:BF:0E:69:8E:CC:49:8D:F5:25:77:EB:F2:E9:3B:9A

Alias name: baltimorecodesigningca
Certificate fingerprints:
  MD5:  90:F5:28:49:56:D1:5D:2C:B0:53:D4:4B:EF:6F:90:22
  SHA1: 30:46:D8:C8:88:FF:69:30:C3:4A:FC:CD:49:27:08:7C:60:56:7B:0D
  SHA256:
 A9:15:45:DB:D2:E1:9C:4C:CD:F9:09:AA:71:90:0D:18:C7:35:1C:89:B3:15:F0:F1:3D:05:C1:3A:8F:FB:46:87

Alias name: gdcatrustauthr5root
Certificate fingerprints:
  MD5:  63:CC:D9:3D:34:35:5C:6F:53:A3:E2:08:70:48:1F:B4
  SHA1: 0F:36:38:5B:81:1A:25:C3:9B:31:4E:83:CA:E9:34:66:70:CC:74:B4
  SHA256:
 BF:FF:8F:D0:44:33:48:7D:6A:8A:A6:0C:1A:29:76:7A:9F:C2:BB:B0:5E:42:0F:71:3A:13:B9:92:89:1D:38:93

Alias name: certinomisrootca
Certificate fingerprints:
  MD5:  14:0A:FD:8D:A8:28:B5:38:69:DB:56:7E:61:22:03:3F
  SHA1: 9D:70:BB:01:A5:A4:A0:18:11:2E:F7:1C:01:B9:32:C5:34:E7:88:A8
  SHA256:
 2A:99:F5:BC:11:74:B7:3C:BB:1D:62:08:84:E0:1C:34:E5:1C:CB:39:78:DA:12:5F:0E:33:26:88:83:BF:41:58

Alias name: verisignclass3publicprimarycertificationauthorityg5
Certificate fingerprints:
  MD5:  CB:17:E4:31:67:3E:E2:09:FE:45:57:93:F3:0A:FA:1C
  SHA1: 4E:B6:D5:78:49:9B:1C:CF:5F:58:1E:AD:56:BE:3D:9B:67:44:A5:E5
  SHA256:
 9A:CF:AB:7E:43:C8:D8:80:D0:6B:26:2A:94:DE:EE:E4:B4:65:99:89:C3:D0:CA:F1:9B:AF:64:05:E4:1A:B7:DF

Alias name: verisignclass3publicprimarycertificationauthorityg4
Certificate fingerprints:
  MD5:  3A:52:E1:E7:FD:6F:3A:E3:6F:F3:6F:99:1B:F9:22:41
```

```
  SHA1: 22:D5:D8:DF:8F:02:31:D1:8D:F7:9D:B7:CF:8A:2D:64:C9:3F:6C:3A
  SHA256:
 69:DD:D7:EA:90:BB:57:C9:3E:13:5D:C8:5E:A6:FC:D5:48:0B:60:32:39:BD:C4:54:FC:75:8B:2A:26:CF:7F:79

Alias name: verisignclass3publicprimarycertificationauthorityg3
Certificate fingerprints:
  MD5:  CD:68:B6:A7:C7:C4:CE:75:E0:1D:4F:57:44:61:92:09
  SHA1: 13:2D:0D:45:53:4B:69:97:CD:B2:D5:C3:39:E2:55:76:60:9B:5C:C6
  SHA256:
 EB:04:CF:5E:B1:F3:9A:FA:76:2F:2B:B1:20:F2:96:CB:A5:20:C1:B9:7D:B1:58:95:65:B8:1C:B9:A1:7B:72:44

Alias name: swisssignsilverg2ca
Certificate fingerprints:
  MD5:  E0:06:A1:C9:7D:CF:C9:FC:0D:C0:56:75:96:D8:62:13
  SHA1: 9B:AA:E5:9F:56:EE:21:CB:43:5A:BE:25:93:DF:A7:F0:40:D1:1D:CB
  SHA256:
 BE:6C:4D:A2:BB:B9:BA:59:B6:F3:93:97:68:37:42:46:C3:C0:05:99:3F:A9:8F:02:0D:1D:ED:BE:D4:8A:81:D5

Alias name: swisssignsilvercag2
Certificate fingerprints:
  MD5:  E0:06:A1:C9:7D:CF:C9:FC:0D:C0:56:75:96:D8:62:13
  SHA1: 9B:AA:E5:9F:56:EE:21:CB:43:5A:BE:25:93:DF:A7:F0:40:D1:1D:CB
  SHA256:
 BE:6C:4D:A2:BB:B9:BA:59:B6:F3:93:97:68:37:42:46:C3:C0:05:99:3F:A9:8F:02:0D:1D:ED:BE:D4:8A:81:D5

Alias name: atostrustedroot2011
Certificate fingerprints:
  MD5:  AE:B9:C4:32:4B:AC:7F:5D:66:CC:77:94:BB:2A:77:56
  SHA1: 2B:B1:F5:3E:55:0C:1D:C5:F1:D4:E6:B7:6A:46:4B:55:06:02:AC:21
  SHA256:
 F3:56:BE:A2:44:B7:A9:1E:B3:5D:53:CA:9A:D7:86:4A:CE:01:8E:2D:35:D5:F8:F9:6D:DF:68:A6:F4:1A:A4:74

Alias name: comodoecccertificationauthority
Certificate fingerprints:
  MD5:  7C:62:FF:74:9D:31:53:5E:68:4A:D5:78:AA:1E:BF:23
  SHA1: 9F:74:4E:9F:2B:4D:BA:EC:0F:31:2C:50:B6:56:3B:8E:2D:93:C3:11
  SHA256:
 17:93:92:7A:06:14:54:97:89:AD:CE:2F:8F:34:F7:F0:B6:6D:0F:3A:E3:A3:B8:4D:21:EC:15:DB:BA:4F:AD:C7

Alias name: securetrustca
Certificate fingerprints:
  MD5:  DC:32:C3:A7:6D:25:57:C7:68:09:9D:EA:2D:A9:A2:D1
  SHA1: 87:82:C6:C3:04:35:3B:CF:D2:96:92:D2:59:3E:7D:44:D9:34:FF:11
  SHA256:
 F1:C1:B5:0A:E5:A2:0D:D8:03:0E:C9:F6:BC:24:82:3D:D3:67:B5:25:57:59:B4:E7:1B:61:FC:E9:F7:37:5D:73

Alias name: soneraclass1ca
Certificate fingerprints:
  MD5:  33:B7:84:F5:5F:27:D7:68:27:DE:14:DE:12:2A:ED:6F
  SHA1: 07:47:22:01:99:CE:74:B9:7C:B0:3D:79:B2:64:A2:C8:55:E9:33:FF
  SHA256:
 CD:80:82:84:CF:74:6F:F2:FD:6E:B5:8A:A1:D5:9C:4A:D4:B3:CA:56:FD:C6:27:4A:89:26:A7:83:5F:32:31:3D

Alias name: cadisigrootr2
Certificate fingerprints:
  MD5:  26:01:FB:D8:27:A7:17:9A:45:54:38:1A:43:01:3B:03
  SHA1: B5:61:EB:EA:A4:DE:E4:25:4B:69:1A:98:A5:57:47:C2:34:C7:D9:71
  SHA256:
 E2:3D:4A:03:6D:7B:70:E9:F5:95:B1:42:20:79:D2:B9:1E:DF:BB:1F:B6:51:A0:63:3E:AA:8A:9D:C5:F8:07:03

Alias name: cadisigrootr1
Certificate fingerprints:
  MD5:  BE:EC:11:93:9A:F5:69:21:BC:D7:C1:C0:67:89:CC:2A
  SHA1: 8E:1C:74:F8:A6:20:B9:E5:8A:F4:61:FA:EC:2B:47:56:51:1A:52:C6
  SHA256:
 F9:6F:23:F4:C3:E7:9C:07:7A:46:98:8D:5A:F5:90:06:76:A0:F0:39:CB:64:5D:D1:75:49:B2:16:C8:24:40:CE
```

```
Alias name: verisignclass3g5ca
Certificate fingerprints:
  MD5:  CB:17:E4:31:67:3E:E2:09:FE:45:57:93:F3:0A:FA:1C
  SHA1: 4E:B6:D5:78:49:9B:1C:CF:5F:58:1E:AD:56:BE:3D:9B:67:44:A5:E5
  SHA256:
 9A:CF:AB:7E:43:C8:D8:80:D0:6B:26:2A:94:DE:EE:E4:B4:65:99:89:C3:D0:CA:F1:9B:AF:64:05:E4:1A:B7:DF

Alias name: utnuserfirsthardwareca
Certificate fingerprints:
  MD5:  4C:56:41:E5:0D:BB:2B:E8:CA:A3:ED:18:08:AD:43:39
  SHA1: 04:83:ED:33:99:AC:36:08:05:87:22:ED:BC:5E:46:00:E3:BE:F9:D7
  SHA256:
 6E:A5:47:41:D0:04:66:7E:ED:1B:48:16:63:4A:A3:A7:9E:6E:4B:96:95:0F:82:79:DA:FC:8D:9B:D8:81:21:37

Alias name: addtrustqualifiedca
Certificate fingerprints:
  MD5:  27:EC:39:47:CD:DA:5A:AF:E2:9A:01:65:21:A9:4C:BB
  SHA1: 4D:23:78:EC:91:95:39:B5:00:7F:75:8F:03:3B:21:1E:C5:4D:8B:CF
  SHA256:
 80:95:21:08:05:DB:4B:BC:35:5E:44:28:D8:FD:6E:C2:CD:E3:AB:5F:B9:7A:99:42:98:8E:B8:F4:DC:D0:60:16

Alias name: verisignclass3g3ca
Certificate fingerprints:
  MD5:  CD:68:B6:A7:C7:C4:CE:75:E0:1D:4F:57:44:61:92:09
  SHA1: 13:2D:0D:45:53:4B:69:97:CD:B2:D5:C3:39:E2:55:76:60:9B:5C:C6
  SHA256:
 EB:04:CF:5E:B1:F3:9A:FA:76:2F:2B:B1:20:F2:96:CB:A5:20:C1:B9:7D:B1:58:95:65:B8:1C:B9:A1:7B:72:44

Alias name: thawtepersonalfreemailca
Certificate fingerprints:
  MD5:  53:4B:1D:17:58:58:1A:30:A1:90:F8:6E:5C:F2:CF:65
  SHA1: E6:18:83:AE:84:CA:C1:C1:CD:52:AD:E8:E9:25:2B:45:A6:4F:B7:E2
  SHA256:
 5B:38:BD:12:9E:83:D5:A0:CA:D2:39:21:08:94:90:D5:0D:4A:AE:37:04:28:F8:DD:FF:FF:FA:4C:15:64:E1:84

Alias name: certplusclass3pprimaryca
Certificate fingerprints:
  MD5:  E1:4B:52:73:D7:1B:DB:93:30:E5:BD:E4:09:6E:BE:FB
  SHA1: 21:6B:2A:29:E6:2A:00:CE:82:01:46:D8:24:41:41:B9:25:11:B2:79
  SHA256:
 CC:C8:94:89:37:1B:AD:11:1C:90:61:9B:EA:24:0A:2E:6D:AD:D9:9F:9F:6E:1D:4D:41:E5:8E:D6:DE:3D:02:85

Alias name: swisssigngoldg2ca
Certificate fingerprints:
  MD5:  24:77:D9:A8:91:D1:3B:FA:88:2D:C2:FF:F8:CD:33:93
  SHA1: D8:C5:38:8A:B7:30:1B:1B:6E:D4:7A:E6:45:25:3A:6F:9F:1A:27:61
  SHA256:
 62:DD:0B:E9:B9:F5:0A:16:3E:A0:F8:E7:5C:05:3B:1E:CA:57:EA:55:C8:68:8F:64:7C:68:81:F2:C8:35:7B:95

Alias name: swisssigngoldcag2
Certificate fingerprints:
  MD5:  24:77:D9:A8:91:D1:3B:FA:88:2D:C2:FF:F8:CD:33:93
  SHA1: D8:C5:38:8A:B7:30:1B:1B:6E:D4:7A:E6:45:25:3A:6F:9F:1A:27:61
  SHA256:
 62:DD:0B:E9:B9:F5:0A:16:3E:A0:F8:E7:5C:05:3B:1E:CA:57:EA:55:C8:68:8F:64:7C:68:81:F2:C8:35:7B:95

Alias name: dtrustrootclass3ca22009
Certificate fingerprints:
  MD5:  CD:E0:25:69:8D:47:AC:9C:89:35:90:F7:FD:51:3D:2F
  SHA1: 58:E8:AB:B0:36:15:33:FB:80:F7:9B:1B:6D:29:D3:FF:8D:5F:00:F0
  SHA256:
 49:E7:A4:42:AC:F0:EA:62:87:05:00:54:B5:25:64:B6:50:E4:F4:9E:42:E3:48:D6:AA:38:E0:39:E9:57:B1:C1

Alias name: acraizfnmtrcm
Certificate fingerprints:
```

```
  MD5:  E2:09:04:B4:D3:BD:D1:A0:14:FD:1A:D2:47:C4:57:1D
  SHA1: EC:50:35:07:B2:15:C4:95:62:19:E2:A8:9A:5B:42:99:2C:4C:2C:20
  SHA256:
 EB:C5:57:0C:29:01:8C:4D:67:B1:AA:12:7B:AF:12:F7:03:B4:61:1E:BC:17:B7:DA:B5:57:38:94:17:9B:93:FA

Alias name: securitycommunicationevrootca1
Certificate fingerprints:
  MD5:  22:2D:A6:01:EA:7C:0A:F7:F0:6C:56:43:3F:77:76:D3
  SHA1: FE:B8:C4:32:DC:F9:76:9A:CE:AE:3D:D8:90:8F:FD:28:86:65:64:7D
  SHA256:
 A2:2D:BA:68:1E:97:37:6E:2D:39:7D:72:8A:AE:3A:9B:62:96:B9:FD:BA:60:BC:2E:11:F6:47:F2:C6:75:FB:37

Alias name: starfieldclass2ca
Certificate fingerprints:
  MD5:  32:4A:4B:BB:C8:63:69:9B:BE:74:9A:C6:DD:1D:46:24
  SHA1: AD:7E:1C:28:B0:64:EF:8F:60:03:40:20:14:C3:D0:E3:37:0E:B5:8A
  SHA256:
 14:65:FA:20:53:97:B8:76:FA:A6:F0:A9:95:8E:55:90:E4:0F:CC:7F:AA:4F:B7:C2:C8:67:75:21:FB:5F:B6:58

Alias name: opentrustrootcag3
Certificate fingerprints:
  MD5:  21:37:B4:17:16:92:7B:67:46:70:A9:96:D7:A8:13:24
  SHA1: 6E:26:64:F3:56:BF:34:55:BF:D1:93:3F:7C:01:DE:D8:13:DA:8A:A6
  SHA256:
 B7:C3:62:31:70:6E:81:07:8C:36:7C:B8:96:19:8F:1E:32:08:DD:92:69:49:DD:8F:57:09:A4:10:F7:5B:62:92

Alias name: opentrustrootcag2
Certificate fingerprints:
  MD5:  57:24:B6:59:24:6B:AE:C8:FE:1C:0C:20:F2:C0:4E:EB
  SHA1: 79:5F:88:60:C5:AB:7C:3D:92:E6:CB:F4:8D:E1:45:CD:11:EF:60:0B
  SHA256:
 27:99:58:29:FE:6A:75:15:C1:BF:E8:48:F9:C4:76:1D:B1:6C:22:59:29:25:7B:F4:0D:08:94:F2:9E:A8:BA:F2

Alias name: buypassclass2rootca
Certificate fingerprints:
  MD5:  46:A7:D2:FE:45:FB:64:5A:A8:59:90:9B:78:44:9B:29
  SHA1: 49:0A:75:74:DE:87:0A:47:FE:58:EE:F6:C7:6B:EB:C6:0B:12:40:99
  SHA256:
 9A:11:40:25:19:7C:5B:B9:5D:94:E6:3D:55:CD:43:79:08:47:B6:46:B2:3C:DF:11:AD:A4:A0:0E:FF:15:FB:48

Alias name: opentrustrootcag1
Certificate fingerprints:
  MD5:  76:00:CC:81:29:CD:55:5E:88:6A:7A:2E:F7:4D:39:DA
  SHA1: 79:91:E8:34:F7:E2:EE:DD:08:95:01:52:E9:55:2D:14:E9:58:D5:7E
  SHA256:
 56:C7:71:28:D9:8C:18:D9:1B:4C:FD:FF:BC:25:EE:91:03:D4:75:8E:A2:AB:AD:82:6A:90:F3:45:7D:46:0E:B4

Alias name: globalsignr2ca
Certificate fingerprints:
  MD5:  94:14:77:7E:3E:5E:FD:8F:30:BD:41:B0:CF:E7:D0:30
  SHA1: 75:E0:AB:B6:13:85:12:27:1C:04:F8:5F:DD:DE:38:E4:B7:24:2E:FE
  SHA256:
 CA:42:DD:41:74:5F:D0:B8:1E:B9:02:36:2C:F9:D8:BF:71:9D:A1:BD:1B:1E:FC:94:6F:5B:4C:99:F4:2C:1B:9E

Alias name: buypassclass3rootca
Certificate fingerprints:
  MD5:  3D:3B:18:9E:2C:64:5A:E8:D5:88:CE:0E:F9:37:C2:EC
  SHA1: DA:FA:F7:FA:66:84:EC:06:8F:14:50:BD:C7:C2:81:A5:BC:A9:64:57
  SHA256:
 ED:F7:EB:BC:A2:7A:2A:38:4D:38:7B:7D:40:10:C6:66:E2:ED:B4:84:3E:4C:29:B4:AE:1D:5B:93:32:E6:B2:4D

Alias name: ecacc
Certificate fingerprints:
  MD5:  EB:F5:9D:29:0D:61:F9:42:1F:7C:C2:BA:6D:E3:15:09
  SHA1: 28:90:3A:63:5B:52:80:FA:E6:77:4C:0B:6D:A7:D6:BA:A6:4A:F2:E8
```

```
  SHA256:
 88:49:7F:01:60:2F:31:54:24:6A:E2:8C:4D:5A:EF:10:F1:D8:7E:BB:76:62:6F:4A:E0:B7:F9:5B:A7:96:87:99

Alias name: epkirootcertificationauthority
Certificate fingerprints:
  MD5:  1B:2E:00:CA:26:06:90:3D:AD:FE:6F:15:68:D3:6B:B3
  SHA1: 67:65:0D:F1:7E:8E:7E:5B:82:40:A4:F4:56:4B:CF:E2:3D:69:C6:F0
  SHA256:
 C0:A6:F4:DC:63:A2:4B:FD:CF:54:EF:2A:6A:08:2A:0A:72:DE:35:80:3E:2F:F5:FF:52:7A:E5:D8:72:06:DF:D5

Alias name: verisignclass1g2ca
Certificate fingerprints:
  MD5:  DB:23:3D:F9:69:FA:4B:B9:95:80:44:73:5E:7D:41:83
  SHA1: 27:3E:E1:24:57:FD:C4:F9:0C:55:E8:2B:56:16:7F:62:F5:32:E5:47
  SHA256:
 34:1D:E9:8B:13:92:AB:F7:F4:AB:90:A9:60:CF:25:D4:BD:6E:C6:5B:9A:51:CE:6E:D0:67:D0:0E:C7:CE:9B:7F

Alias name: certigna
Certificate fingerprints:
  MD5:  AB:57:A6:5B:7D:42:82:19:B5:D8:58:26:28:5E:FD:FF
  SHA1: B1:2E:13:63:45:86:A4:6F:1A:B2:60:68:37:58:2D:C4:AC:FD:94:97
  SHA256:
 E3:B6:A2:DB:2E:D7:CE:48:84:2F:7A:C5:32:41:C7:B7:1D:54:14:4B:FB:40:C1:1F:3F:1D:0B:42:F5:EE:A1:2D

Alias name: camerfirmaglobalchambersignroot
Certificate fingerprints:
  MD5:  C5:E6:7B:BF:06:D0:4F:43:ED:C4:7A:65:8A:FB:6B:19
  SHA1: 33:9B:6B:14:50:24:9B:55:7A:01:87:72:84:D9:E0:2F:C3:D2:D8:E9
  SHA256:
 EF:3C:B4:17:FC:8E:BF:6F:97:87:6C:9E:4E:CE:39:DE:1E:A5:FE:64:91:41:D1:02:8B:7D:11:C0:B2:29:8C:ED

Alias name: cfcaevroot
Certificate fingerprints:
  MD5:  74:E1:B6:ED:26:7A:7A:44:30:33:94:AB:7B:27:81:30
  SHA1: E2:B8:29:4B:55:84:AB:6B:58:C2:90:46:6C:AC:3F:B8:39:8F:84:83
  SHA256:
 5C:C3:D7:8E:4E:1D:5E:45:54:7A:04:E6:87:3E:64:F9:0C:F9:53:6D:1C:CC:2E:F8:00:F3:55:C4:C5:FD:70:FD

Alias name: soneraclass2rootca
Certificate fingerprints:
  MD5:  A3:EC:75:0F:2E:88:DF:FA:48:01:4E:0B:5C:48:6F:FB
  SHA1: 37:F7:6D:E6:07:7C:90:C5:B1:3E:93:1A:B7:41:10:B4:F2:E4:9A:27
  SHA256:
 79:08:B4:03:14:C1:38:10:0B:51:8D:07:35:80:7F:FB:FC:F8:51:8A:00:95:33:71:05:BA:38:6B:15:3D:D9:27

Alias name: certumtrustednetworkca
Certificate fingerprints:
  MD5:  D5:E9:81:40:C5:18:69:FC:46:2C:89:75:62:0F:AA:78
  SHA1: 07:E0:32:E0:20:B7:2C:3F:19:2F:06:28:A2:59:3A:19:A7:0F:06:9E
  SHA256:
 5C:58:46:8D:55:F5:8E:49:7E:74:39:82:D2:B5:00:10:B6:D1:65:37:4A:CF:83:A7:D4:A3:2D:B7:68:C4:40:8E

Alias name: securitycommunicationrootca2
Certificate fingerprints:
  MD5:  6C:39:7D:A4:0E:55:59:B2:3F:D6:41:B1:12:50:DE:43
  SHA1: 5F:3B:8C:F2:F8:10:B3:7D:78:B4:CE:EC:19:19:C3:73:34:B9:C7:74
  SHA256:
 51:3B:2C:EC:B8:10:D4:CD:E5:DD:85:39:1A:DF:C6:C2:DD:60:D8:7B:B7:36:D2:B5:21:48:4A:A4:7A:0E:BE:F6

Alias name: globalsigneccrootcar5
Certificate fingerprints:
  MD5:  9F:AD:3B:1C:02:1E:8A:BA:17:74:38:81:0C:A2:BC:08
  SHA1: 1F:24:C6:30:CD:A4:18:EF:20:69:FF:AD:4F:DD:5F:46:3A:1B:69:AA
  SHA256:
 17:9F:BC:14:8A:3D:D0:0F:D2:4E:A1:34:58:CC:43:BF:A7:F5:9C:81:82:D7:83:A5:13:F6:EB:EC:10:0C:89:24
```

```
Alias name: globalsigneccrootcar4
Certificate fingerprints:
  MD5:  20:F0:27:68:D1:7E:A0:9D:0E:E6:2A:CA:DF:5C:89:8E
  SHA1: 69:69:56:2E:40:80:F4:24:A1:E7:19:9F:14:BA:F3:EE:58:AB:6A:BB
  SHA256:
 BE:C9:49:11:C2:95:56:76:DB:6C:0A:55:09:86:D7:6E:3B:A0:05:66:7C:44:2C:97:62:B4:FB:B7:73:DE:22:8C

Alias name: chambersofcommerceroot2008
Certificate fingerprints:
  MD5:  5E:80:9E:84:5A:0E:65:0B:17:02:F3:55:18:2A:3E:D7
  SHA1: 78:6A:74:AC:76:AB:14:7F:9C:6A:30:50:BA:9E:A8:7E:FE:9A:CE:3C
  SHA256:
 06:3E:4A:FA:C4:91:DF:D3:32:F3:08:9B:85:42:E9:46:17:D8:93:D7:FE:94:4E:10:A7:93:7E:E2:9D:96:93:C0

Alias name: pscprocert
Certificate fingerprints:
  MD5:  E6:24:E9:12:01:AE:0C:DE:8E:85:C4:CE:A3:12:DD:EC
  SHA1: 70:C1:8D:74:B4:28:81:0A:E4:FD:A5:75:D7:01:9F:99:B0:3D:50:74
  SHA256:
 3C:FC:3C:14:D1:F6:84:FF:17:E3:8C:43:CA:44:0C:00:B9:67:EC:93:3E:8B:FE:06:4C:A1:D7:2C:90:F2:AD:B0

Alias name: thawteprimaryrootcag3
Certificate fingerprints:
  MD5:  FB:1B:5D:43:8A:94:CD:44:C6:76:F2:43:4B:47:E7:31
  SHA1: F1:8B:53:8D:1B:E9:03:B6:A6:F0:56:43:5B:17:15:89:CA:F3:6B:F2
  SHA256:
 4B:03:F4:58:07:AD:70:F2:1B:FC:2C:AE:71:C9:FD:E4:60:4C:06:4C:F5:FF:B6:86:BA:E5:DB:AA:D7:FD:D3:4C

Alias name: quovadisrootca
Certificate fingerprints:
  MD5:  27:DE:36:FE:72:B7:00:03:00:9D:F4:F0:1E:6C:04:24
  SHA1: DE:3F:40:BD:50:93:D3:9B:6C:60:F6:DA:BC:07:62:01:00:89:76:C9
  SHA256:
 A4:5E:DE:3B:BB:F0:9C:8A:E1:5C:72:EF:C0:72:68:D6:93:A2:1C:99:6F:D5:1E:67:CA:07:94:60:FD:6D:88:73

Alias name: thawteprimaryrootcag2
Certificate fingerprints:
  MD5:  74:9D:EA:60:24:C4:FD:22:53:3E:CC:3A:72:D9:29:4F
  SHA1: AA:DB:BC:22:23:8F:C4:01:A1:27:BB:38:DD:F4:1D:DB:08:9E:F0:12
  SHA256:
 A4:31:0D:50:AF:18:A6:44:71:90:37:2A:86:AF:AF:8B:95:1F:FB:43:1D:83:7F:1E:56:88:B4:59:71:ED:15:57

Alias name: deprecateditsecca
Certificate fingerprints:
  MD5:  A5:96:0C:F6:B5:AB:27:E5:01:C6:00:88:9E:60:33:E5
  SHA1: 12:12:0B:03:0E:15:14:54:F4:DD:B3:F5:DE:13:6E:83:5A:29:72:9D
  SHA256:
 9A:59:DA:86:24:1A:FD:BA:A3:39:FA:9C:FD:21:6A:0B:06:69:4D:E3:7E:37:52:6B:BE:63:C8:BC:83:74:2E:CB

Alias name: usertrustrsacertificationauthority
Certificate fingerprints:
  MD5:  1B:FE:69:D1:91:B7:19:33:A3:72:A8:0F:E1:55:E5:B5
  SHA1: 2B:8F:1B:57:33:0D:BB:A2:D0:7A:6C:51:F7:0E:E9:0D:DA:B9:AD:8E
  SHA256:
 E7:93:C9:B0:2F:D8:AA:13:E2:1C:31:22:8A:CC:B0:81:19:64:3B:74:9C:89:89:64:B1:74:6D:46:C3:D4:CB:D2

Alias name: entrustrootcag2
Certificate fingerprints:
  MD5:  4B:E2:C9:91:96:65:0C:F4:0E:5A:93:92:A0:0A:FE:B2
  SHA1: 8C:F4:27:FD:79:0C:3A:D1:66:06:8D:E8:1E:57:EF:BB:93:22:72:D4
  SHA256:
 43:DF:57:74:B0:3E:7F:EF:5F:E4:0D:93:1A:7B:ED:F1:BB:2E:6B:42:73:8C:4E:6D:38:41:10:3D:3A:A7:F3:39

Alias name: networksolutionscertificateauthority
Certificate fingerprints:
  MD5:  D3:F3:A6:16:C0:FA:6B:1D:59:B1:2D:96:4D:0E:11:2E
```

```
  SHA1: 74:F8:A3:C3:EF:E7:B3:90:06:4B:83:90:3C:21:64:60:20:E5:DF:CE
  SHA256:
 15:F0:BA:00:A3:AC:7A:F3:AC:88:4C:07:2B:10:11:A0:77:BD:77:C0:97:F4:01:64:B2:F8:59:8A:BD:83:86:0C

Alias name: trustcenterclass4caii
Certificate fingerprints:
  MD5:  9D:FB:F9:AC:ED:89:33:22:F4:28:48:83:25:23:5B:E0
  SHA1: A6:9A:91:FD:05:7F:13:6A:42:63:0B:B1:76:0D:2D:51:12:0C:16:50
  SHA256:
 32:66:96:7E:59:CD:68:00:8D:9D:D3:20:81:11:85:C7:04:20:5E:8D:95:FD:D8:4F:1C:7B:31:1E:67:04:FC:32

Alias name: oistewisekeyglobalrootgaca
Certificate fingerprints:
  MD5:  BC:6C:51:33:A7:E9:D3:66:63:54:15:72:1B:21:92:93
  SHA1: 59:22:A1:E1:5A:EA:16:35:21:F8:98:39:6A:46:46:B0:44:1B:0F:A9
  SHA256:
 41:C9:23:86:6A:B4:CA:D6:B7:AD:57:80:81:58:2E:02:07:97:A6:CB:DF:4F:FF:78:CE:83:96:B3:89:37:D7:F5

Alias name: verisignuniversalrootcertificationauthority
Certificate fingerprints:
  MD5:  8E:AD:B5:01:AA:4D:81:E4:8C:1D:D1:E1:14:00:95:19
  SHA1: 36:79:CA:35:66:87:72:30:4D:30:A5:FB:87:3B:0F:A7:7B:B7:0D:54
  SHA256:
 23:99:56:11:27:A5:71:25:DE:8C:EF:EA:61:0D:DF:2F:A0:78:B5:C8:06:7F:4E:82:82:90:BF:B8:60:E8:4B:3C

Alias name: ttelesecglobalrootclass3ca
Certificate fingerprints:
  MD5:  CA:FB:40:A8:4E:39:92:8A:1D:FE:8E:2F:C4:27:EA:EF
  SHA1: 55:A6:72:3E:CB:F2:EC:CD:C3:23:74:70:19:9D:2A:BE:11:E3:81:D1
  SHA256:
 FD:73:DA:D3:1C:64:4F:F1:B4:3B:EF:0C:CD:DA:96:71:0B:9C:D9:87:5E:CA:7E:31:70:7A:F3:E9:6D:52:2B:BD

Alias name: starfieldservicesrootg2ca
Certificate fingerprints:
  MD5:  17:35:74:AF:7B:61:1C:EB:F4:F9:3C:E2:EE:40:F9:A2
  SHA1: 92:5A:8F:8D:2C:6D:04:E0:66:5F:59:6A:FF:22:D8:63:E8:25:6F:3F
  SHA256:
 56:8D:69:05:A2:C8:87:08:A4:B3:02:51:90:ED:CF:ED:B1:97:4A:60:6A:13:C6:E5:29:0F:CB:2A:E6:3E:DA:B5

Alias name: addtrustexternalroot
Certificate fingerprints:
  MD5:  1D:35:54:04:85:78:B0:3F:42:42:4D:BF:20:73:0A:3F
  SHA1: 02:FA:F3:E2:91:43:54:68:60:78:57:69:4D:F5:E4:5B:68:85:18:68
  SHA256:
 68:7F:A4:51:38:22:78:FF:F0:C8:B1:1F:8D:43:D5:76:67:1C:6E:B2:BC:EA:B4:13:FB:83:D9:65:D0:6D:2F:F2

Alias name: turktrustelektroniksertifikahizmetsaglayicisih5
Certificate fingerprints:
  MD5:  DA:70:8E:F0:22:DF:93:26:F6:5F:9F:D3:15:06:52:4E
  SHA1: C4:18:F6:4D:46:D1:DF:00:3D:27:30:13:72:43:A9:12:11:C6:75:FB
  SHA256:
 49:35:1B:90:34:44:C1:85:CC:DC:5C:69:3D:24:D8:55:5C:B2:08:D6:A8:14:13:07:69:9F:4A:F0:63:19:9D:78

Alias name: camerfirmachambersca
Certificate fingerprints:
  MD5:  5E:80:9E:84:5A:0E:65:0B:17:02:F3:55:18:2A:3E:D7
  SHA1: 78:6A:74:AC:76:AB:14:7F:9C:6A:30:50:BA:9E:A8:7E:FE:9A:CE:3C
  SHA256:
 06:3E:4A:FA:C4:91:DF:D3:32:F3:08:9B:85:42:E9:46:17:D8:93:D7:FE:94:4E:10:A7:93:7E:E2:9D:96:93:C0

Alias name: certsignrootca
Certificate fingerprints:
  MD5:  18:98:C0:D6:E9:3A:FC:F9:B0:F5:0C:F7:4B:01:44:17
  SHA1: FA:B7:EE:36:97:26:62:FB:2D:B0:2A:F6:BF:03:FD:E8:7C:4B:2F:9B
  SHA256:
 EA:A9:62:C4:FA:4A:6B:AF:EB:E4:15:19:6D:35:1C:CD:88:8D:4F:53:F3:FA:8A:E6:D7:C4:66:A9:4E:60:42:BB
```

```
Alias name: verisignuniversalrootca
Certificate fingerprints:
  MD5:  8E:AD:B5:01:AA:4D:81:E4:8C:1D:D1:E1:14:00:95:19
  SHA1: 36:79:CA:35:66:87:72:30:4D:30:A5:FB:87:3B:0F:A7:7B:B7:0D:54
  SHA256:
 23:99:56:11:27:A5:71:25:DE:8C:EF:EA:61:0D:DF:2F:A0:78:B5:C8:06:7F:4E:82:82:90:BF:B8:60:E8:4B:3C

Alias name: geotrustuniversalca
Certificate fingerprints:
  MD5:  92:65:58:8B:A2:1A:31:72:73:68:5C:B4:A5:7A:07:48
  SHA1: E6:21:F3:35:43:79:05:9A:4B:68:30:9D:8A:2F:74:22:15:87:EC:79
  SHA256:
 A0:45:9B:9F:63:B2:25:59:F5:FA:5D:4C:6D:B3:F9:F7:2F:F1:93:42:03:35:78:F0:73:BF:1D:1B:46:CB:B9:12

Alias name: luxtrustglobalroot2
Certificate fingerprints:
  MD5:  B2:E1:09:00:61:AF:F7:F1:91:6F:C4:AD:8D:5E:3B:7C
  SHA1: 1E:0E:56:19:0A:D1:8B:25:98:B2:04:44:FF:66:8A:04:17:99:5F:3F
  SHA256:
 54:45:5F:71:29:C2:0B:14:47:C4:18:F9:97:16:8F:24:C5:8F:C5:02:3B:F5:DA:5B:E2:EB:6E:1D:D8:90:2E:D5

Alias name: twcaglobalrootca
Certificate fingerprints:
  MD5:  F9:03:7E:CF:E6:9E:3C:73:7A:2A:90:07:69:FF:2B:96
  SHA1: 9C:BB:48:53:F6:A4:F6:D3:52:A4:E8:32:52:55:60:13:F5:AD:AF:65
  SHA256:
 59:76:90:07:F7:68:5D:0F:CD:50:87:2F:9F:95:D5:75:5A:5B:2B:45:7D:81:F3:69:2B:61:0A:98:67:2F:0E:1B

Alias name: tubitakkamusmsslkoksertifikasisurum1
Certificate fingerprints:
  MD5:  DC:00:81:DC:69:2F:3E:2F:B0:3B:F6:3D:5A:91:8E:49
  SHA1: 31:43:64:9B:EC:CE:27:EC:ED:3A:3F:0B:8F:0D:E4:E8:91:DD:EE:CA
  SHA256:
 46:ED:C3:68:90:46:D5:3A:45:3F:B3:10:4A:B8:0D:CA:EC:65:8B:26:60:EA:16:29:DD:7E:86:79:90:64:87:16

Alias name: affirmtrustnetworkingca
Certificate fingerprints:
  MD5:  42:65:CA:BE:01:9A:9A:4C:A9:8C:41:49:CD:C0:D5:7F
  SHA1: 29:36:21:02:8B:20:ED:02:F5:66:C5:32:D1:D6:ED:90:9F:45:00:2F
  SHA256:
 0A:81:EC:5A:92:97:77:F1:45:90:4A:F3:8D:5D:50:9F:66:B5:E2:C5:8F:CD:B5:31:05:8B:0E:17:F3:F0:B4:1B

Alias name: affirmtrustcommercialca
Certificate fingerprints:
  MD5:  82:92:BA:5B:EF:CD:8A:6F:A6:3D:55:F9:84:F6:D6:B7
  SHA1: F9:B5:B6:32:45:5F:9C:BE:EC:57:5F:80:DC:E9:6E:2C:C7:B2:78:B7
  SHA256:
 03:76:AB:1D:54:C5:F9:80:3C:E4:B2:E2:01:A0:EE:7E:EF:7B:57:B6:36:E8:A9:3C:9B:8D:48:60:C9:6F:5F:A7

Alias name: godaddyrootcertificateauthorityg2
Certificate fingerprints:
  MD5:  80:3A:BC:22:C1:E6:FB:8D:9B:3B:27:4A:32:1B:9A:01
  SHA1: 47:BE:AB:C9:22:EA:E8:0E:78:78:34:62:A7:9F:45:C2:54:FD:E6:8B
  SHA256:
 45:14:0B:32:47:EB:9C:C8:C5:B4:F0:D7:B5:30:91:F7:32:92:08:9E:6E:5A:63:E2:74:9D:D3:AC:A9:19:8E:DA

Alias name: starfieldrootg2ca
Certificate fingerprints:
  MD5:  D6:39:81:C6:52:7E:96:69:FC:FC:CA:66:ED:05:F2:96
  SHA1: B5:1C:06:7C:EE:2B:0C:3D:F8:55:AB:2D:92:F4:FE:39:D4:E7:0F:0E
  SHA256:
 2C:E1:CB:0B:F9:D2:F9:E1:02:99:3F:BE:21:51:52:C3:B2:DD:0C:AB:DE:1C:68:E5:31:9B:83:91:54:DB:B7:F5

Alias name: dtrustrootclass3ca2ev2009
Certificate fingerprints:
```

```
  MD5:  AA:C6:43:2C:5E:2D:CD:C4:34:C0:50:4F:11:02:4F:B6
  SHA1: 96:C9:1B:0B:95:B4:10:98:42:FA:D0:D8:22:79:FE:60:FA:B9:16:83
  SHA256:
 EE:C5:49:6B:98:8C:E9:86:25:B9:34:09:2E:EC:29:08:BE:D0:B0:F3:16:C2:D4:73:0C:84:EA:F1:F3:D3:48:81

Alias name: buypassclass3ca
Certificate fingerprints:
  MD5:  3D:3B:18:9E:2C:64:5A:E8:D5:88:CE:0E:F9:37:C2:EC
  SHA1: DA:FA:F7:FA:66:84:EC:06:8F:14:50:BD:C7:C2:81:A5:BC:A9:64:57
  SHA256:
 ED:F7:EB:BC:A2:7A:2A:38:4D:38:7B:7D:40:10:C6:66:E2:ED:B4:84:3E:4C:29:B4:AE:1D:5B:93:32:E6:B2:4D

Alias name: verisignclass2g3ca
Certificate fingerprints:
  MD5:  F8:BE:C4:63:22:C9:A8:46:74:8B:B8:1D:1E:4A:2B:F6
  SHA1: 61:EF:43:D7:7F:CA:D4:61:51:BC:98:E0:C3:59:12:AF:9F:EB:63:11
  SHA256:
 92:A9:D9:83:3F:E1:94:4D:B3:66:E8:BF:AE:7A:95:B6:48:0C:2D:6C:6C:2A:1B:E6:5D:42:36:B6:08:FC:A1:BB

Alias name: digicerttrustedrootg4
Certificate fingerprints:
  MD5:  78:F2:FC:AA:60:1F:2F:B4:EB:C9:37:BA:53:2E:75:49
  SHA1: DD:FB:16:CD:49:31:C9:73:A2:03:7D:3F:C8:3A:4D:7D:77:5D:05:E4
  SHA256:
 55:2F:7B:DC:F1:A7:AF:9E:6C:E6:72:01:7F:4F:12:AB:F7:72:40:C7:8E:76:1A:C2:03:D1:D9:D2:0A:C8:99:88

Alias name: quovadisrootca2g3
Certificate fingerprints:
  MD5:  AF:0C:86:6E:BF:40:2D:7F:0B:3E:12:50:BA:12:3D:06
  SHA1: 09:3C:61:F3:8B:8B:DC:7D:55:DF:75:38:02:05:00:E1:25:F5:C8:36
  SHA256:
 8F:E4:FB:0A:F9:3A:4D:0D:67:DB:0B:EB:B2:3E:37:C7:1B:F3:25:DC:BC:DD:24:0E:A0:4D:AF:58:B4:7E:18:40

Alias name: geotrustprimarycertificationauthorityg3
Certificate fingerprints:
  MD5:  B5:E8:34:36:C9:10:44:58:48:70:6D:2E:83:D4:B8:05
  SHA1: 03:9E:ED:B8:0B:E7:A0:3C:69:53:89:3B:20:D2:D9:32:3A:4C:2A:FD
  SHA256:
 B4:78:B8:12:25:0D:F8:78:63:5C:2A:A7:EC:7D:15:5E:AA:62:5E:E8:29:16:E2:CD:29:43:61:88:6C:D1:FB:D4

Alias name: geotrustprimarycertificationauthorityg2
Certificate fingerprints:
  MD5:  01:5E:D8:6B:BD:6F:3D:8E:A1:31:F8:12:E0:98:73:6A
  SHA1: 8D:17:84:D5:37:F3:03:7D:EC:70:FE:57:8B:51:9A:99:E6:10:D7:B0
  SHA256:
 5E:DB:7A:C4:3B:82:A0:6A:87:61:E8:D7:BE:49:79:EB:F2:61:1F:7D:D7:9B:F9:1C:1C:6B:56:6A:21:9E:D7:66

Alias name: godaddyclass2ca
Certificate fingerprints:
  MD5:  91:DE:06:25:AB:DA:FD:32:17:0C:BB:25:17:2A:84:67
  SHA1: 27:96:BA:E6:3F:18:01:E2:77:26:1B:A0:D7:77:70:02:8F:20:EE:E4
  SHA256:
 C3:84:6B:F2:4B:9E:93:CA:64:27:4C:0E:C6:7C:1E:CC:5E:02:4F:FC:AC:D2:D7:40:19:35:0E:81:FE:54:6A:E4

Alias name: trustcoreca1
Certificate fingerprints:
  MD5:  27:92:23:1D:0A:F5:40:7C:E9:E6:6B:9D:D8:F5:E7:6C
  SHA1: 58:D1:DF:95:95:67:6B:63:C0:F0:5B:1C:17:4D:8B:84:0B:C8:78:BD
  SHA256:
 5A:88:5D:B1:9C:01:D9:12:C5:75:93:88:93:8C:AF:BB:DF:03:1A:B2:D4:8E:91:EE:15:58:9B:42:97:1D:03:9C

Alias name: hellenicacademicandresearchinstitutionseccrootca2015
Certificate fingerprints:
  MD5:  81:E5:B4:17:EB:C2:F5:E1:4B:0D:41:7B:49:92:FE:EF
  SHA1: 9F:F1:71:8D:92:D5:9A:F3:7D:74:97:B4:BC:6F:84:68:0B:BA:B6:66
```

```
  SHA256:
 44:B5:45:AA:8A:25:E6:5A:73:CA:15:DC:27:FC:36:D2:4C:1C:B9:95:3A:06:65:39:B1:15:82:DC:48:7B:48:33

Alias name: utnuserfirstobjectca
Certificate fingerprints:
  MD5:  A7:F2:E4:16:06:41:11:50:30:6B:9C:E3:B4:9C:B0:C9
  SHA1: E1:2D:FB:4B:41:D7:D9:C3:2B:30:51:4B:AC:1D:81:D8:38:5E:2D:46
  SHA256:
 6F:FF:78:E4:00:A7:0C:11:01:1C:D8:59:77:C4:59:FB:5A:F9:6A:3D:F0:54:08:20:D0:F4:B8:60:78:75:E5:8F

Alias name: ttelesecglobalrootclass3
Certificate fingerprints:
  MD5:  CA:FB:40:A8:4E:39:92:8A:1D:FE:8E:2F:C4:27:EA:EF
  SHA1: 55:A6:72:3E:CB:F2:EC:CD:C3:23:74:70:19:9D:2A:BE:11:E3:81:D1
  SHA256:
 FD:73:DA:D3:1C:64:4F:F1:B4:3B:EF:0C:CD:DA:96:71:0B:9C:D9:87:5E:CA:7E:31:70:7A:F3:E9:6D:52:2B:BD

Alias name: ttelesecglobalrootclass2
Certificate fingerprints:
  MD5:  2B:9B:9E:E4:7B:6C:1F:00:72:1A:CC:C1:77:79:DF:6A
  SHA1: 59:0D:2D:7D:88:4F:40:2E:61:7E:A5:62:32:17:65:CF:17:D8:94:E9
  SHA256:
 91:E2:F5:78:8D:58:10:EB:A7:BA:58:73:7D:E1:54:8A:8E:CA:CD:01:45:98:BC:0B:14:3E:04:1B:17:05:25:52

Alias name: addtrustclass1ca
Certificate fingerprints:
  MD5:  1E:42:95:02:33:92:6B:B9:5F:C0:7F:DA:D6:B2:4B:FC
  SHA1: CC:AB:0E:A0:4C:23:01:D6:69:7B:DD:37:9F:CD:12:EB:24:E3:94:9D
  SHA256:
 8C:72:09:27:9A:C0:4E:27:5E:16:D0:7F:D3:B7:75:E8:01:54:B5:96:80:46:E3:1F:52:DD:25:76:63:24:E9:A7

Alias name: amzninternalrootca
Certificate fingerprints:
  MD5:  08:09:73:AC:E0:78:41:7C:0A:26:33:51:E8:CF:E6:60
  SHA1: A7:B7:F6:15:8A:FF:1E:C8:85:13:38:BC:93:EB:A2:AB:A4:09:EF:06
  SHA256:
 0E:DE:63:C1:DC:7A:8E:11:F1:AB:BC:05:4F:59:EE:49:9D:62:9A:2F:DE:9C:A7:16:32:A2:64:29:3E:8B:66:AA

Alias name: starfieldrootcertificateauthorityg2
Certificate fingerprints:
  MD5:  D6:39:81:C6:52:7E:96:69:FC:FC:CA:66:ED:05:F2:96
  SHA1: B5:1C:06:7C:EE:2B:0C:3D:F8:55:AB:2D:92:F4:FE:39:D4:E7:0F:0E
  SHA256:
 2C:E1:CB:0B:F9:D2:F9:E1:02:99:3F:BE:21:51:52:C3:B2:DD:0C:AB:DE:1C:68:E5:31:9B:83:91:54:DB:B7:F5

Alias name: camerfirmachambersignca
Certificate fingerprints:
  MD5:  9E:80:FF:78:01:0C:2E:C1:36:BD:FE:96:90:6E:08:F3
  SHA1: 4A:BD:EE:EC:95:0D:35:9C:89:AE:C7:52:A1:2C:5B:29:F6:D6:AA:0C
  SHA256:
 13:63:35:43:93:34:A7:69:80:16:A0:D3:24:DE:72:28:4E:07:9D:7B:52:20:BB:8F:BD:74:78:16:EE:BE:BA:CA

Alias name: secomscrootca2
Certificate fingerprints:
  MD5:  6C:39:7D:A4:0E:55:59:B2:3F:D6:41:B1:12:50:DE:43
  SHA1: 5F:3B:8C:F2:F8:10:B3:7D:78:B4:CE:EC:19:19:C3:73:34:B9:C7:74
  SHA256:
 51:3B:2C:EC:B8:10:D4:CD:E5:DD:85:39:1A:DF:C6:C2:DD:60:D8:7B:B7:36:D2:B5:21:48:4A:A4:7A:0E:BE:F6

Alias name: entrustevca
Certificate fingerprints:
  MD5:  D6:A5:C3:ED:5D:DD:3E:00:C1:3D:87:92:1F:1D:3F:E4
  SHA1: B3:1E:B1:B7:40:E3:6C:84:02:DA:DC:37:D4:4D:F5:D4:67:49:52:F9
  SHA256:
 73:C1:76:43:4F:1B:C6:D5:AD:F4:5B:0E:76:E7:27:28:7C:8D:E5:76:16:C1:E6:E6:14:1A:2B:2C:BC:7D:8E:4C
```

```
Alias name: secomscrootca1
Certificate fingerprints:
  MD5:  F1:BC:63:6A:54:E0:B5:27:F5:CD:E7:1A:E3:4D:6E:4A
  SHA1: 36:B1:2B:49:F9:81:9E:D7:4C:9E:BC:38:0F:C6:56:8F:5D:AC:B2:F7
  SHA256:
 E7:5E:72:ED:9F:56:0E:EC:6E:B4:80:00:73:A4:3F:C3:AD:19:19:5A:39:22:82:01:78:95:97:4A:99:02:6B:6C

Alias name: affirmtrustcommercial
Certificate fingerprints:
  MD5:  82:92:BA:5B:EF:CD:8A:6F:A6:3D:55:F9:84:F6:D6:B7
  SHA1: F9:B5:B6:32:45:5F:9C:BE:EC:57:5F:80:DC:E9:6E:2C:C7:B2:78:B7
  SHA256:
 03:76:AB:1D:54:C5:F9:80:3C:E4:B2:E2:01:A0:EE:7E:EF:7B:57:B6:36:E8:A9:3C:9B:8D:48:60:C9:6F:5F:A7

Alias name: digicertassuredidrootg3
Certificate fingerprints:
  MD5:  7C:7F:65:31:0C:81:DF:8D:BA:3E:99:E2:5C:AD:6E:FB
  SHA1: F5:17:A2:4F:9A:48:C6:C9:F8:A2:00:26:9F:DC:0F:48:2C:AB:30:89
  SHA256:
 7E:37:CB:8B:4C:47:09:0C:AB:36:55:1B:A6:F4:5D:B8:40:68:0F:BA:16:6A:95:2D:B1:00:71:7F:43:05:3F:C2

Alias name: affirmtrustnetworking
Certificate fingerprints:
  MD5:  42:65:CA:BE:01:9A:9A:4C:A9:8C:41:49:CD:C0:D5:7F
  SHA1: 29:36:21:02:8B:20:ED:02:F5:66:C5:32:D1:D6:ED:90:9F:45:00:2F
  SHA256:
 0A:81:EC:5A:92:97:77:F1:45:90:4A:F3:8D:5D:50:9F:66:B5:E2:C5:8F:CD:B5:31:05:8B:0E:17:F3:F0:B4:1B

Alias name: izenpecom
Certificate fingerprints:
  MD5:  A6:B0:CD:85:80:DA:5C:50:34:A3:39:90:2F:55:67:73
  SHA1: 2F:78:3D:25:52:18:A7:4A:65:39:71:B5:2C:A2:9C:45:15:6F:E9:19
  SHA256:
 25:30:CC:8E:98:32:15:02:BA:D9:6F:9B:1F:BA:1B:09:9E:2D:29:9E:0F:45:48:BB:91:4F:36:3B:C0:D4:53:1F

Alias name: amazon-ca-g4-legacy
Certificate fingerprints:
  MD5:  6C:E5:BD:67:A4:4F:E3:FD:C2:4C:46:E6:06:5B:6D:55
  SHA1: EA:E7:DE:F9:0A:BE:9F:0B:68:CE:B7:24:0D:80:74:03:BF:6E:B1:6E
  SHA256:
 CD:72:C4:7F:B4:AD:28:A4:67:2B:E1:86:47:D4:40:E9:3B:16:2D:95:DB:3C:2F:94:BB:81:D9:09:F7:91:24:5E

Alias name: digicertassuredidrootg2
Certificate fingerprints:
  MD5:  92:38:B9:F8:63:24:82:65:2C:57:33:E6:FE:81:8F:9D
  SHA1: A1:4B:48:D9:43:EE:0A:0E:40:90:4F:3C:E0:A4:C0:91:93:51:5D:3F
  SHA256:
 7D:05:EB:B6:82:33:9F:8C:94:51:EE:09:4E:EB:FE:FA:79:53:A1:14:ED:B2:F4:49:49:45:2F:AB:7D:2F:C1:85

Alias name: comodoaaaservicesroot
Certificate fingerprints:
  MD5:  49:79:04:B0:EB:87:19:AC:47:B0:BC:11:51:9B:74:D0
  SHA1: D1:EB:23:A4:6D:17:D6:8F:D9:25:64:C2:F1:F1:60:17:64:D8:E3:49
  SHA256:
 D7:A7:A0:FB:5D:7E:27:31:D7:71:E9:48:4E:BC:DE:F7:1D:5F:0C:3E:0A:29:48:78:2B:C8:3E:E0:EA:69:9E:F4

Alias name: entrustnetpremium2048secureserverca
Certificate fingerprints:
  MD5:  EE:29:31:BC:32:7E:9A:E6:E8:B5:F7:51:B4:34:71:90
  SHA1: 50:30:06:09:1D:97:D4:F5:AE:39:F7:CB:E7:92:7D:7D:65:2D:34:31
  SHA256:
 6D:C4:71:72:E0:1C:BC:B0:BF:62:58:0D:89:5F:E2:B8:AC:9A:D4:F8:73:80:1E:0C:10:B9:C8:37:D2:1E:B1:77

Alias name: trustcorrootcertca2
Certificate fingerprints:
  MD5:  A2:E1:F8:18:0B:BA:45:D5:C7:41:2A:BB:37:52:45:64
```

```
  SHA1: B8:BE:6D:CB:56:F1:55:B9:63:D4:12:CA:4E:06:34:C7:94:B2:1C:C0
  SHA256:
 07:53:E9:40:37:8C:1B:D5:E3:83:6E:39:5D:AE:A5:CB:83:9E:50:46:F1:BD:0E:AE:19:51:CF:10:FE:C7:C9:65

Alias name: entrust2048ca
Certificate fingerprints:
  MD5:  EE:29:31:BC:32:7E:9A:E6:E8:B5:F7:51:B4:34:71:90
  SHA1: 50:30:06:09:1D:97:D4:F5:AE:39:F7:CB:E7:92:7D:7D:65:2D:34:31
  SHA256:
 6D:C4:71:72:E0:1C:BC:B0:BF:62:58:0D:89:5F:E2:B8:AC:9A:D4:F8:73:80:1E:0C:10:B9:C8:37:D2:1E:B1:77

Alias name: trustcorrootcertca1
Certificate fingerprints:
  MD5:  6E:85:F1:DC:1A:00:D3:22:D5:B2:B2:AC:6B:37:05:45
  SHA1: FF:BD:CD:E7:82:C8:43:5E:3C:6F:26:86:5C:CA:A8:3A:45:5B:C3:0A
  SHA256:
 D4:0E:9C:86:CD:8F:E4:68:C1:77:69:59:F4:9E:A7:74:FA:54:86:84:B6:C4:06:F3:90:92:61:F4:DC:E2:57:5C

Alias name: baltimorecybertrustroot
Certificate fingerprints:
  MD5:  AC:B6:94:A5:9C:17:E0:D7:91:52:9B:B1:97:06:A6:E4
  SHA1: D4:DE:20:D0:5E:66:FC:53:FE:1A:50:88:2C:78:DB:28:52:CA:E4:74
  SHA256:
 16:AF:57:A9:F6:76:B0:AB:12:60:95:AA:5E:BA:DE:F2:2A:B3:11:19:D6:44:AC:95:CD:4B:93:DB:F3:F2:6A:EB

Alias name: eecertificationcentrerootca
Certificate fingerprints:
  MD5:  43:5E:88:D4:7D:1A:4A:7E:FD:84:2E:52:EB:01:D4:6F
  SHA1: C9:A8:B9:E7:55:80:5E:58:E3:53:77:A7:25:EB:AF:C3:7B:27:CC:D7
  SHA256:
 3E:84:BA:43:42:90:85:16:E7:75:73:C0:99:2F:09:79:CA:08:4E:46:85:68:1F:F1:95:CC:BA:8A:22:9B:8A:76

Alias name: dstacescax6
Certificate fingerprints:
  MD5:  21:D8:4C:82:2B:99:09:33:A2:EB:14:24:8D:8E:5F:E8
  SHA1: 40:54:DA:6F:1C:3F:40:74:AC:ED:0F:EC:CD:DB:79:D1:53:FB:90:1D
  SHA256:
 76:7C:95:5A:76:41:2C:89:AF:68:8E:90:A1:C7:0F:55:6C:FD:6B:60:25:DB:EA:10:41:6D:7E:B6:83:1F:8C:40

Alias name: comodocertificationauthority
Certificate fingerprints:
  MD5:  5C:48:DC:F7:42:72:EC:56:94:6D:1C:CC:71:35:80:75
  SHA1: 66:31:BF:9E:F7:4F:9E:B6:C9:D5:A6:0C:BA:6A:BE:D1:F7:BD:EF:7B
  SHA256:
 0C:2C:D6:3D:F7:80:6F:A3:99:ED:E8:09:11:6B:57:5B:F8:79:89:F0:65:18:F9:80:8C:86:05:03:17:8B:AF:66

Alias name: thawteserverca
Certificate fingerprints:
  MD5:  EE:FE:61:69:65:6E:F8:9C:C6:2A:F4:D7:2B:63:EF:A2
  SHA1: 9F:AD:91:A6:CE:6A:C6:C5:00:47:C4:4E:C9:D4:A5:0D:92:D8:49:79
  SHA256:
 87:C6:78:BF:B8:B2:5F:38:F7:E9:7B:33:69:56:BB:CF:14:4B:BA:CA:A5:36:47:E6:1A:23:25:BC:10:55:31:6B

Alias name: secomvalicertclass1ca
Certificate fingerprints:
  MD5:  65:58:AB:15:AD:57:6C:1E:A8:A7:B5:69:AC:BF:FF:EB
  SHA1: E5:DF:74:3C:B6:01:C4:9B:98:43:DC:AB:8C:E8:6A:81:10:9F:E4:8E
  SHA256:
 F4:C1:49:55:1A:30:13:A3:5B:C7:BF:FE:17:A7:F3:44:9B:C1:AB:5B:5A:0A:E7:4B:06:C2:3B:90:00:4C:01:04

Alias name: godaddyrootg2ca
Certificate fingerprints:
  MD5:  80:3A:BC:22:C1:E6:FB:8D:9B:3B:27:4A:32:1B:9A:01
  SHA1: 47:BE:AB:C9:22:EA:E8:0E:78:78:34:62:A7:9F:45:C2:54:FD:E6:8B
  SHA256:
 45:14:0B:32:47:EB:9C:C8:C5:B4:F0:D7:B5:30:91:F7:32:92:08:9E:6E:5A:63:E2:74:9D:D3:AC:A9:19:8E:DA
```

```
Alias name: globalchambersignroot2008
Certificate fingerprints:
  MD5:  9E:80:FF:78:01:0C:2E:C1:36:BD:FE:96:90:6E:08:F3
  SHA1: 4A:BD:EE:EC:95:0D:35:9C:89:AE:C7:52:A1:2C:5B:29:F6:D6:AA:0C
  SHA256:
 13:63:35:43:93:34:A7:69:80:16:A0:D3:24:DE:72:28:4E:07:9D:7B:52:20:BB:8F:BD:74:78:16:EE:BE:BA:CA

Alias name: equifaxsecureebusinessca1
Certificate fingerprints:
  MD5:  14:C0:08:E5:A3:85:03:A3:BE:78:E9:67:4F:27:CA:EE
  SHA1: AE:E6:3D:70:E3:76:FB:C7:3A:EB:B0:A1:C1:D4:C4:7A:A7:40:B3:F4
  SHA256:
 2E:3A:2B:B5:11:25:05:83:6C:A8:96:8B:E2:CB:37:27:CE:9B:56:84:5C:6E:E9:8E:91:85:10:4A:FB:9A:F5:96

Alias name: quovadisrootca3
Certificate fingerprints:
  MD5:  31:85:3C:62:94:97:63:B9:AA:FD:89:4E:AF:6F:E0:CF
  SHA1: 1F:49:14:F7:D8:74:95:1D:DD:AE:02:C0:BE:FD:3A:2D:82:75:51:85
  SHA256:
 18:F1:FC:7F:20:5D:F8:AD:DD:EB:7F:E0:07:DD:57:E3:AF:37:5A:9C:4D:8D:73:54:6B:F4:F1:FE:D1:E1:8D:35

Alias name: usertrustecccertificationauthority
Certificate fingerprints:
  MD5:  FA:68:BC:D9:B5:7F:AD:FD:C9:1D:06:83:28:CC:24:C1
  SHA1: D1:CB:CA:5D:B2:D5:2A:7F:69:3B:67:4D:E5:F0:5A:1D:0C:95:7D:F0
  SHA256:
 4F:F4:60:D5:4B:9C:86:DA:BF:BC:FC:57:12:E0:40:0D:2B:ED:3F:BC:4D:4F:BD:AA:86:E0:6A:DC:D2:A9:AD:7A

Alias name: quovadisrootca2
Certificate fingerprints:
  MD5:  5E:39:7B:DD:F8:BA:EC:82:E9:AC:62:BA:0C:54:00:2B
  SHA1: CA:3A:FB:CF:12:40:36:4B:44:B2:16:20:88:80:48:39:19:93:7C:F7
  SHA256:
 85:A0:DD:7D:D7:20:AD:B7:FF:05:F8:3D:54:2B:20:9D:C7:FF:45:28:F7:D6:77:B1:83:89:FE:A5:E5:C4:9E:86

Alias name: soneraclass2ca
Certificate fingerprints:
  MD5:  A3:EC:75:0F:2E:88:DF:FA:48:01:4E:0B:5C:48:6F:FB
  SHA1: 37:F7:6D:E6:07:7C:90:C5:B1:3E:93:1A:B7:41:10:B4:F2:E4:9A:27
  SHA256:
 79:08:B4:03:14:C1:38:10:0B:51:8D:07:35:80:7F:FB:FC:F8:51:8A:00:95:33:71:05:BA:38:6B:15:3D:D9:27

Alias name: twcarootcertificationauthority
Certificate fingerprints:
  MD5:  AA:08:8F:F6:F9:7B:B7:F2:B1:A7:1E:9B:EA:EA:BD:79
  SHA1: CF:9E:87:6D:D3:EB:FC:42:26:97:A3:B5:A3:7A:A0:76:A9:06:23:48
  SHA256:
 BF:D8:8F:E1:10:1C:41:AE:3E:80:1B:F8:BE:56:35:0E:E9:BA:D1:A6:B9:BD:51:5E:DC:5C:6D:5B:87:11:AC:44

Alias name: baltimorecybertrustca
Certificate fingerprints:
  MD5:  AC:B6:94:A5:9C:17:E0:D7:91:52:9B:B1:97:06:A6:E4
  SHA1: D4:DE:20:D0:5E:66:FC:53:FE:1A:50:88:2C:78:DB:28:52:CA:E4:74
  SHA256:
 16:AF:57:A9:F6:76:B0:AB:12:60:95:AA:5E:BA:DE:F2:2A:B3:11:19:D6:44:AC:95:CD:4B:93:DB:F3:F2:6A:EB

Alias name: cia-crt-g3-01-ca
Certificate fingerprints:
  MD5:  E3:66:DD:D6:A0:D5:40:8F:FF:29:E2:C0:CB:6E:62:1A
  SHA1: 2B:EE:2C:BA:A3:1D:B5:FE:60:40:41:95:08:ED:46:82:39:4D:ED:E2
  SHA256:
 20:48:AD:4C:EC:90:7F:FA:4A:15:D4:CE:45:E3:C8:E4:2C:EA:78:33:DC:C7:D3:40:48:FC:60:47:27:42:99:EC

Alias name: entrustrootcertificationauthorityg2
Certificate fingerprints:
```

```
  MD5:  4B:E2:C9:91:96:65:0C:F4:0E:5A:93:92:A0:0A:FE:B2
  SHA1: 8C:F4:27:FD:79:0C:3A:D1:66:06:8D:E8:1E:57:EF:BB:93:22:72:D4
  SHA256:
 43:DF:57:74:B0:3E:7F:EF:5F:E4:0D:93:1A:7B:ED:F1:BB:2E:6B:42:73:8C:4E:6D:38:41:10:3D:3A:A7:F3:39

Alias name: verisignclass3g4ca
Certificate fingerprints:
  MD5:  3A:52:E1:E7:FD:6F:3A:E3:6F:F3:6F:99:1B:F9:22:41
  SHA1: 22:D5:D8:DF:8F:02:31:D1:8D:F7:9D:B7:CF:8A:2D:64:C9:3F:6C:3A
  SHA256:
 69:DD:D7:EA:90:BB:57:C9:3E:13:5D:C8:5E:A6:FC:D5:48:0B:60:32:39:BD:C4:54:FC:75:8B:2A:26:CF:7F:79

Alias name: xrampglobalcaroot
Certificate fingerprints:
  MD5:  A1:0B:44:B3:CA:10:D8:00:6E:9D:0F:D8:0F:92:0A:D1
  SHA1: B8:01:86:D1:EB:9C:86:A5:41:04:CF:30:54:F3:4C:52:B7:E5:58:C6
  SHA256:
 CE:CD:DC:90:50:99:D8:DA:DF:C5:B1:D2:09:B7:37:CB:E2:C1:8C:FB:2C:10:C0:FF:0B:CF:0D:32:86:FC:1A:A2

Alias name: identrustcommercialrootca1
Certificate fingerprints:
  MD5:  B3:3E:77:73:75:EE:A0:D3:E3:7E:49:63:49:59:BB:C7
  SHA1: DF:71:7E:AA:4A:D9:4E:C9:55:84:99:60:2D:48:DE:5F:BC:F0:3A:25
  SHA256:
 5D:56:49:9B:E4:D2:E0:8B:CF:CA:D0:8A:3E:38:72:3D:50:50:3B:DE:70:69:48:E4:2F:55:60:30:19:E5:28:AE

Alias name: camerfirmachamberscommerceca
Certificate fingerprints:
  MD5:  B0:01:EE:14:D9:AF:29:18:94:76:8E:F1:69:33:2A:84
  SHA1: 6E:3A:55:A4:19:0C:19:5C:93:84:3C:C0:DB:72:2E:31:30:61:F0:B1
  SHA256:
 0C:25:8A:12:A5:67:4A:EF:25:F2:8B:A7:DC:FA:EC:EE:A3:48:E5:41:E6:F5:CC:4E:E6:3B:71:B3:61:60:6A:C3

Alias name: verisignclass3g2ca
Certificate fingerprints:
  MD5:  A2:33:9B:4C:74:78:73:D4:6C:E7:C1:F3:8D:CB:5C:E9
  SHA1: 85:37:1C:A6:E5:50:14:3D:CE:28:03:47:1B:DE:3A:09:E8:F8:77:0F
  SHA256:
 83:CE:3C:12:29:68:8A:59:3D:48:5F:81:97:3C:0F:91:95:43:1E:DA:37:CC:5E:36:43:0E:79:C7:A8:88:63:8B

Alias name: deutschetelekomrootca2
Certificate fingerprints:
  MD5:  74:01:4A:91:B1:08:C4:58:CE:47:CD:F0:DD:11:53:08
  SHA1: 85:A4:08:C0:9C:19:3E:5D:51:58:7D:CD:D6:13:30:FD:8C:DE:37:BF
  SHA256:
 B6:19:1A:50:D0:C3:97:7F:7D:A9:9B:CD:AA:C8:6A:22:7D:AE:B9:67:9E:C7:0B:A3:B0:C9:D9:22:71:C1:70:D3

Alias name: certumca
Certificate fingerprints:
  MD5:  2C:8F:9F:66:1D:18:90:B1:47:26:9D:8E:86:82:8C:A9
  SHA1: 62:52:DC:40:F7:11:43:A2:2F:DE:9E:F7:34:8E:06:42:51:B1:81:18
  SHA256:
 D8:E0:FE:BC:1D:B2:E3:8D:00:94:0F:37:D2:7D:41:34:4D:99:3E:73:4B:99:D5:65:6D:97:78:D4:D8:14:36:24

Alias name: cybertrustglobalroot
Certificate fingerprints:
  MD5:  72:E4:4A:87:E3:69:40:80:77:EA:BC:E3:F4:FF:F0:E1
  SHA1: 5F:43:E5:B1:BF:F8:78:8C:AC:1C:C7:CA:4A:9A:C6:22:2B:CC:34:C6
  SHA256:
 96:0A:DF:00:63:E9:63:56:75:0C:29:65:DD:0A:08:67:DA:0B:9C:BD:6E:77:71:4A:EA:FB:23:49:AB:39:3D:A3

Alias name: globalsignrootca
Certificate fingerprints:
  MD5:  3E:45:52:15:09:51:92:E1:B7:5D:37:9F:B1:87:29:8A
  SHA1: B1:BC:96:8B:D4:F4:9D:62:2A:A8:9A:81:F2:15:01:52:A4:1D:82:9C
```

```
  SHA256:
 EB:D4:10:40:E4:BB:3E:C7:42:C9:E3:81:D3:1E:F2:A4:1A:48:B6:68:5C:96:E7:CE:F3:C1:DF:6C:D4:33:1C:99

Alias name: secomevrootca1
Certificate fingerprints:
  MD5:  22:2D:A6:01:EA:7C:0A:F7:F0:6C:56:43:3F:77:76:D3
  SHA1: FE:B8:C4:32:DC:F9:76:9A:CE:AE:3D:D8:90:8F:FD:28:86:65:64:7D
  SHA256:
 A2:2D:BA:68:1E:97:37:6E:2D:39:7D:72:8A:AE:3A:9B:62:96:B9:FD:BA:60:BC:2E:11:F6:47:F2:C6:75:FB:37

Alias name: globalsignr3ca
Certificate fingerprints:
  MD5:  C5:DF:B8:49:CA:05:13:55:EE:2D:BA:1A:C3:3E:B0:28
  SHA1: D6:9B:56:11:48:F0:1C:77:C5:45:78:C1:09:26:DF:5B:85:69:76:AD
  SHA256:
 CB:B5:22:D7:B7:F1:27:AD:6A:01:13:86:5B:DF:1C:D4:10:2E:7D:07:59:AF:63:5A:7C:F4:72:0D:C9:63:C5:3B

Alias name: staatdernederlandenrootcag3
Certificate fingerprints:
  MD5:  0B:46:67:07:DB:10:2F:19:8C:35:50:60:D1:0B:F4:37
  SHA1: D8:EB:6B:41:51:92:59:E0:F3:E7:85:00:C0:3D:B6:88:97:C9:EE:FC
  SHA256:
 3C:4F:B0:B9:5A:B8:B3:00:32:F4:32:B8:6F:53:5F:E1:72:C1:85:D0:FD:39:86:58:37:CF:36:18:7F:A6:F4:28

Alias name: staatdernederlandenrootcag2
Certificate fingerprints:
  MD5:  7C:A5:0F:F8:5B:9A:7D:6D:30:AE:54:5A:E3:42:A2:8A
  SHA1: 59:AF:82:79:91:86:C7:B4:75:07:CB:CF:03:57:46:EB:04:DD:B7:16
  SHA256:
 66:8C:83:94:7D:A6:3B:72:4B:EC:E1:74:3C:31:A0:E6:AE:D0:DB:8E:C5:B3:1B:E3:77:BB:78:4F:91:B6:71:6F

Alias name: aolrootca2
Certificate fingerprints:
  MD5:  D6:ED:3C:CA:E2:66:0F:AF:10:43:0D:77:9B:04:09:BF
  SHA1: 85:B5:FF:67:9B:0C:79:96:1F:C8:6E:44:22:00:46:13:DB:17:92:84
  SHA256:
 7D:3B:46:5A:60:14:E5:26:C0:AF:FC:EE:21:27:D2:31:17:27:AD:81:1C:26:84:2D:00:6A:F3:73:06:CC:80:BD

Alias name: dstrootcax3
Certificate fingerprints:
  MD5:  41:03:52:DC:0F:F7:50:1B:16:F0:02:8E:BA:6F:45:C5
  SHA1: DA:C9:02:4F:54:D8:F6:DF:94:93:5F:B1:73:26:38:CA:6A:D7:7C:13
  SHA256:
 06:87:26:03:31:A7:24:03:D9:09:F1:05:E6:9B:CF:0D:32:E1:BD:24:93:FF:C6:D9:20:6D:11:BC:D6:77:07:39

Alias name: trustcenteruniversalcai
Certificate fingerprints:
  MD5:  45:E1:A5:72:C5:A9:36:64:40:9E:F5:E4:58:84:67:8C
  SHA1: 6B:2F:34:AD:89:58:BE:62:FD:B0:6B:5C:CE:BB:9D:D9:4F:4E:39:F3
  SHA256:
 EB:F3:C0:2A:87:89:B1:FB:7D:51:19:95:D6:63:B7:29:06:D9:13:CE:0D:5E:10:56:8A:8A:77:E2:58:61:67:E7

Alias name: aolrootca1
Certificate fingerprints:
  MD5:  14:F1:08:AD:9D:FA:64:E2:89:E7:1C:CF:A8:AD:7D:5E
  SHA1: 39:21:C1:15:C1:5D:0E:CA:5C:CB:5B:C4:F0:7D:21:D8:05:0B:56:6A
  SHA256:
 77:40:73:12:C6:3A:15:3D:5B:C0:0B:4E:51:75:9C:DF:DA:C2:37:DC:2A:33:B6:79:46:E9:8E:9B:FA:68:0A:E3

Alias name: affirmtrustpremiumecc
Certificate fingerprints:
  MD5:  64:B0:09:55:CF:B1:D5:99:E2:BE:13:AB:A6:5D:EA:4D
  SHA1: B8:23:6B:00:2F:1D:16:86:53:01:55:6C:11:A4:37:CA:EB:FF:C3:BB
  SHA256:
 BD:71:FD:F6:DA:97:E4:CF:62:D1:64:7A:DD:25:81:B0:7D:79:AD:F8:39:7E:B4:EC:BA:9C:5E:84:88:82:14:23
```

```
Alias name: microseceszignorootca2009
Certificate fingerprints:
  MD5:  F8:49:F4:03:BC:44:2D:83:BE:48:69:7D:29:64:FC:B1
  SHA1: 89:DF:74:FE:5C:F4:0F:4A:80:F9:E3:37:7D:54:DA:91:E1:01:31:8E
  SHA256:
 3C:5F:81:FE:A5:FA:B8:2C:64:BF:A2:EA:EC:AF:CD:E8:E0:77:FC:86:20:A7:CA:E5:37:16:3D:F3:6E:DB:F3:78

Alias name: verisignclass1g3ca
Certificate fingerprints:
  MD5:  B1:47:BC:18:57:D1:18:A0:78:2D:EC:71:E8:2A:95:73
  SHA1: 20:42:85:DC:F7:EB:76:41:95:57:8E:13:6B:D4:B7:D1:E9:8E:46:A5
  SHA256:
 CB:B5:AF:18:5E:94:2A:24:02:F9:EA:CB:C0:ED:5B:B8:76:EE:A3:C1:22:36:23:D0:04:47:E4:F3:BA:55:4B:65

Alias name: certplusrootcag2
Certificate fingerprints:
  MD5:  A7:EE:C4:78:2D:1B:EE:2D:B9:29:CE:D6:A7:96:32:31
  SHA1: 4F:65:8E:1F:E9:06:D8:28:02:E9:54:47:41:C9:54:25:5D:69:CC:1A
  SHA256:
 6C:C0:50:41:E6:44:5E:74:69:6C:4C:FB:C9:F8:0F:54:3B:7E:AB:BB:44:B4:CE:6F:78:7C:6A:99:71:C4:2F:17

Alias name: certplusrootcag1
Certificate fingerprints:
  MD5:  7F:09:9C:F7:D9:B9:5C:69:69:56:D5:37:3E:14:0D:42
  SHA1: 22:FD:D0:B7:FD:A2:4E:0D:AC:49:2C:A0:AC:A6:7B:6A:1F:E3:F7:66
  SHA256:
 15:2A:40:2B:FC:DF:2C:D5:48:05:4D:22:75:B3:9C:7F:CA:3E:C0:97:80:78:B0:F0:EA:76:E5:61:A6:C7:43:3E

Alias name: addtrustexternalca
Certificate fingerprints:
  MD5:  1D:35:54:04:85:78:B0:3F:42:42:4D:BF:20:73:0A:3F
  SHA1: 02:FA:F3:E2:91:43:54:68:60:78:57:69:4D:F5:E4:5B:68:85:18:68
  SHA256:
 68:7F:A4:51:38:22:78:FF:F0:C8:B1:1F:8D:43:D5:76:67:1C:6E:B2:BC:EA:B4:13:FB:83:D9:65:D0:6D:2F:F2

Alias name: entrustrootcertificationauthority
Certificate fingerprints:
  MD5:  D6:A5:C3:ED:5D:DD:3E:00:C1:3D:87:92:1F:1D:3F:E4
  SHA1: B3:1E:B1:B7:40:E3:6C:84:02:DA:DC:37:D4:4D:F5:D4:67:49:52:F9
  SHA256:
 73:C1:76:43:4F:1B:C6:D5:AD:F4:5B:0E:76:E7:27:28:7C:8D:E5:76:16:C1:E6:E6:14:1A:2B:2C:BC:7D:8E:4C

Alias name: verisignclass3ca
Certificate fingerprints:
  MD5:  EF:5A:F1:33:EF:F1:CD:BB:51:02:EE:12:14:4B:96:C4
  SHA1: A1:DB:63:93:91:6F:17:E4:18:55:09:40:04:15:C7:02:40:B0:AE:6B
  SHA256:
 A4:B6:B3:99:6F:C2:F3:06:B3:FD:86:81:BD:63:41:3D:8C:50:09:CC:4F:A3:29:C2:CC:F0:E2:FA:1B:14:03:05

Alias name: digicertassuredidrootca
Certificate fingerprints:
  MD5:  87:CE:0B:7B:2A:0E:49:00:E1:58:71:9B:37:A8:93:72
  SHA1: 05:63:B8:63:0D:62:D7:5A:BB:C8:AB:1E:4B:DF:B5:A8:99:B2:4D:43
  SHA256:
 3E:90:99:B5:01:5E:8F:48:6C:00:BC:EA:9D:11:1E:E7:21:FA:BA:35:5A:89:BC:F1:DF:69:56:1E:3D:C6:32:5C

Alias name: globalsignrootcar3
Certificate fingerprints:
  MD5:  C5:DF:B8:49:CA:05:13:55:EE:2D:BA:1A:C3:3E:B0:28
  SHA1: D6:9B:56:11:48:F0:1C:77:C5:45:78:C1:09:26:DF:5B:85:69:76:AD
  SHA256:
 CB:B5:22:D7:B7:F1:27:AD:6A:01:13:86:5B:DF:1C:D4:10:2E:7D:07:59:AF:63:5A:7C:F4:72:0D:C9:63:C5:3B

Alias name: globalsignrootcar2
Certificate fingerprints:
  MD5:  94:14:77:7E:3E:5E:FD:8F:30:BD:41:B0:CF:E7:D0:30
```

```
  SHA1: 75:E0:AB:B6:13:85:12:27:1C:04:F8:5F:DD:DE:38:E4:B7:24:2E:FE
  SHA256:
 CA:42:DD:41:74:5F:D0:B8:1E:B9:02:36:2C:F9:D8:BF:71:9D:A1:BD:1B:1E:FC:94:6F:5B:4C:99:F4:2C:1B:9E

Alias name: verisignclass1ca
Certificate fingerprints:
  MD5:  86:AC:DE:2B:C5:6D:C3:D9:8C:28:88:D3:8D:16:13:1E
  SHA1: CE:6A:64:A3:09:E4:2F:BB:D9:85:1C:45:3E:64:09:EA:E8:7D:60:F1
  SHA256:
 51:84:7C:8C:BD:2E:9A:72:C9:1E:29:2D:2A:E2:47:D7:DE:1E:3F:D2:70:54:7A:20:EF:7D:61:0F:38:B8:84:2C

Alias name: thawtepremiumserverca
Certificate fingerprints:
  MD5:  A6:6B:60:90:23:9B:3F:2D:BB:98:6F:D6:A7:19:0D:46
  SHA1: E0:AB:05:94:20:72:54:93:05:60:62:02:36:70:F7:CD:2E:FC:66:66
  SHA256:
 3F:9F:27:D5:83:20:4B:9E:09:C8:A3:D2:06:6C:4B:57:D3:A2:47:9C:36:93:65:08:80:50:56:98:10:5D:BC:E9

Alias name: verisigntsaca
Certificate fingerprints:
  MD5:  F2:89:95:6E:4D:05:F0:F1:A7:21:55:7D:46:11:BA:47
  SHA1: 20:CE:B1:F0:F5:1C:0E:19:A9:F3:8D:B1:AA:8E:03:8C:AA:7A:C7:01
  SHA256:
 CB:6B:05:D9:E8:E5:7C:D8:82:B1:0B:4D:B7:0D:E4:BB:1D:E4:2B:A4:8A:7B:D0:31:8B:63:5B:F6:E7:78:1A:9D

Alias name: thawteprimaryrootca
Certificate fingerprints:
  MD5:  8C:CA:DC:0B:22:CE:F5:BE:72:AC:41:1A:11:A8:D8:12
  SHA1: 91:C6:D6:EE:3E:8A:C8:63:84:E5:48:C2:99:29:5C:75:6C:81:7B:81
  SHA256:
 8D:72:2F:81:A9:C1:13:C0:79:1D:F1:36:A2:96:6D:B2:6C:95:0A:97:1D:B4:6B:41:99:F4:EA:54:B7:8B:FB:9F

Alias name: visaecommerceroot
Certificate fingerprints:
  MD5:  FC:11:B8:D8:08:93:30:00:6D:23:F9:7E:EB:52:1E:02
  SHA1: 70:17:9B:86:8C:00:A4:FA:60:91:52:22:3F:9F:3E:32:BD:E0:05:62
  SHA256:
 69:FA:C9:BD:55:FB:0A:C7:8D:53:BB:EE:5C:F1:D5:97:98:9F:D0:AA:AB:20:A2:51:51:BD:F1:73:3E:E7:D1:22

Alias name: digicertglobalrootg3
Certificate fingerprints:
  MD5:  F5:5D:A4:50:A5:FB:28:7E:1E:0F:0D:CC:96:57:56:CA
  SHA1: 7E:04:DE:89:6A:3E:66:6D:00:E6:87:D3:3F:FA:D9:3B:E8:3D:34:9E
  SHA256:
 31:AD:66:48:F8:10:41:38:C7:38:F3:9E:A4:32:01:33:39:3E:3A:18:CC:02:29:6E:F9:7C:2A:C9:EF:67:31:D0

Alias name: xrampglobalca
Certificate fingerprints:
  MD5:  A1:0B:44:B3:CA:10:D8:00:6E:9D:0F:D8:0F:92:0A:D1
  SHA1: B8:01:86:D1:EB:9C:86:A5:41:04:CF:30:54:F3:4C:52:B7:E5:58:C6
  SHA256:
 CE:CD:DC:90:50:99:D8:DA:DF:C5:B1:D2:09:B7:37:CB:E2:C1:8C:FB:2C:10:C0:FF:0B:CF:0D:32:86:FC:1A:A2

Alias name: digicertglobalrootg2
Certificate fingerprints:
  MD5:  E4:A6:8A:C8:54:AC:52:42:46:0A:FD:72:48:1B:2A:44
  SHA1: DF:3C:24:F9:BF:D6:66:76:1B:26:80:73:FE:06:D1:CC:8D:4F:82:A4
  SHA256:
 CB:3C:CB:B7:60:31:E5:E0:13:8F:8D:D3:9A:23:F9:DE:47:FF:C3:5E:43:C1:14:4C:EA:27:D4:6A:5A:B1:CB:5F

Alias name: valicertclass2ca
Certificate fingerprints:
  MD5:  A9:23:75:9B:BA:49:36:6E:31:C2:DB:F2:E7:66:BA:87
  SHA1: 31:7A:2A:D0:7F:2B:33:5E:F5:A1:C3:4E:4B:57:E8:B7:D8:F1:FC:A6
  SHA256:
 58:D0:17:27:9C:D4:DC:63:AB:DD:B1:96:A6:C9:90:6C:30:C4:E0:87:83:EA:E8:C1:60:99:54:D6:93:55:59:6B
```

```
Alias name: geotrustprimaryca
Certificate fingerprints:
  MD5:  02:26:C3:01:5E:08:30:37:43:A9:D0:7D:CF:37:E6:BF
  SHA1: 32:3C:11:8E:1B:F7:B8:B6:52:54:E2:E2:10:0D:D6:02:90:37:F0:96
  SHA256:
 37:D5:10:06:C5:12:EA:AB:62:64:21:F1:EC:8C:92:01:3F:C5:F8:2A:E9:8E:E5:33:EB:46:19:B8:DE:B4:D0:6C

Alias name: netlockaranyclassgoldfotanusitvany
Certificate fingerprints:
  MD5:  C5:A1:B7:FF:73:DD:D6:D7:34:32:18:DF:FC:3C:AD:88
  SHA1: 06:08:3F:59:3F:15:A1:04:A0:69:A4:6B:A9:03:D0:06:B7:97:09:91
  SHA256:
 6C:61:DA:C3:A2:DE:F0:31:50:6B:E0:36:D2:A6:FE:40:19:94:FB:D1:3D:F9:C8:D4:66:59:92:74:C4:46:EC:98

Alias name: geotrustglobalca
Certificate fingerprints:
  MD5:  F7:75:AB:29:FB:51:4E:B7:77:5E:FF:05:3C:99:8E:F5
  SHA1: DE:28:F4:A4:FF:E5:B9:2F:A3:C5:03:D1:A3:49:A7:F9:96:2A:82:12
  SHA256:
 FF:85:6A:2D:25:1D:CD:88:D3:66:56:F4:50:12:67:98:CF:AB:AA:DE:40:79:9C:72:2D:E4:D2:B5:DB:36:A7:3A

Alias name: oistewisekeyglobalrootgbca
Certificate fingerprints:
  MD5:  A4:EB:B9:61:28:2E:B7:2F:98:B0:35:26:90:99:51:1D
  SHA1: 0F:F9:40:76:18:D3:D7:6A:4B:98:F0:A8:35:9E:0C:FD:27:AC:CC:ED
  SHA256:
 6B:9C:08:E8:6E:B0:F7:67:CF:AD:65:CD:98:B6:21:49:E5:49:4A:67:F5:84:5E:7B:D1:ED:01:9F:27:B8:6B:D6

Alias name: certumtrustednetworkca2
Certificate fingerprints:
  MD5:  6D:46:9E:D9:25:6D:08:23:5B:5E:74:7D:1E:27:DB:F2
  SHA1: D3:DD:48:3E:2B:BF:4C:05:E8:AF:10:F5:FA:76:26:CF:D3:DC:30:92
  SHA256:
 B6:76:F2:ED:DA:E8:77:5C:D3:6C:B0:F6:3C:D1:D4:60:39:61:F4:9E:62:65:BA:01:3A:2F:03:07:B6:D0:B8:04

Alias name: starfieldservicesrootcertificateauthorityg2
Certificate fingerprints:
  MD5:  17:35:74:AF:7B:61:1C:EB:F4:F9:3C:E2:EE:40:F9:A2
  SHA1: 92:5A:8F:8D:2C:6D:04:E0:66:5F:59:6A:FF:22:D8:63:E8:25:6F:3F
  SHA256:
 56:8D:69:05:A2:C8:87:08:A4:B3:02:51:90:ED:CF:ED:B1:97:4A:60:6A:13:C6:E5:29:0F:CB:2A:E6:3E:DA:B5

Alias name: comodorsacertificationauthority
Certificate fingerprints:
  MD5:  1B:31:B0:71:40:36:CC:14:36:91:AD:C4:3E:FD:EC:18
  SHA1: AF:E5:D2:44:A8:D1:19:42:30:FF:47:9F:E2:F8:97:BB:CD:7A:8C:B4
  SHA256:
 52:F0:E1:C4:E5:8E:C6:29:29:1B:60:31:7F:07:46:71:B8:5D:7E:A8:0D:5B:07:27:34:63:53:4B:32:B4:02:34

Alias name: comodoaaaca
Certificate fingerprints:
  MD5:  49:79:04:B0:EB:87:19:AC:47:B0:BC:11:51:9B:74:D0
  SHA1: D1:EB:23:A4:6D:17:D6:8F:D9:25:64:C2:F1:F1:60:17:64:D8:E3:49
  SHA256:
 D7:A7:A0:FB:5D:7E:27:31:D7:71:E9:48:4E:BC:DE:F7:1D:5F:0C:3E:0A:29:48:78:2B:C8:3E:E0:EA:69:9E:F4

Alias name: identrustpublicsectorrootca1
Certificate fingerprints:
  MD5:  37:06:A5:B0:FC:89:9D:BA:F4:6B:8C:1A:64:CD:D5:BA
  SHA1: BA:29:41:60:77:98:3F:F4:F3:EF:F2:31:05:3B:2E:EA:6D:4D:45:FD
  SHA256:
 30:D0:89:5A:9A:44:8A:26:20:91:63:55:22:D1:F5:20:10:B5:86:7A:CA:E1:2C:78:EF:95:8F:D4:F4:38:9F:2F

Alias name: certplusclass2primaryca
Certificate fingerprints:
```

```
  MD5:  88:2C:8C:52:B8:A2:3C:F3:F7:BB:03:EA:AE:AC:42:0B
  SHA1: 74:20:74:41:72:9C:DD:92:EC:79:31:D8:23:10:8D:C2:81:92:E2:BB
  SHA256:
 0F:99:3C:8A:EF:97:BA:AF:56:87:14:0E:D5:9A:D1:82:1B:B4:AF:AC:F0:AA:9A:58:B5:D5:7A:33:8A:3A:FB:CB

Alias name: ttelesecglobalrootclass2ca
Certificate fingerprints:
  MD5:  2B:9B:9E:E4:7B:6C:1F:00:72:1A:CC:C1:77:79:DF:6A
  SHA1: 59:0D:2D:7D:88:4F:40:2E:61:7E:A5:62:32:17:65:CF:17:D8:94:E9
  SHA256:
 91:E2:F5:78:8D:58:10:EB:A7:BA:58:73:7D:E1:54:8A:8E:CA:CD:01:45:98:BC:0B:14:3E:04:1B:17:05:25:52

Alias name: accvraiz1
Certificate fingerprints:
  MD5:  D0:A0:5A:EE:05:B6:09:94:21:A1:7D:F1:B2:29:82:02
  SHA1: 93:05:7A:88:15:C6:4F:CE:88:2F:FA:91:16:52:28:78:BC:53:64:17
  SHA256:
 9A:6E:C0:12:E1:A7:DA:9D:BE:34:19:4D:47:8A:D7:C0:DB:18:22:FB:07:1D:F1:29:81:49:6E:D1:04:38:41:13

Alias name: digicerthighassuranceevrootca
Certificate fingerprints:
  MD5:  D4:74:DE:57:5C:39:B2:D3:9C:85:83:C5:C0:65:49:8A
  SHA1: 5F:B7:EE:06:33:E2:59:DB:AD:0C:4C:9A:E6:D3:8F:1A:61:C7:DC:25
  SHA256:
 74:31:E5:F4:C3:C1:CE:46:90:77:4F:0B:61:E0:54:40:88:3B:A9:A0:1E:D0:0B:A6:AB:D7:80:6E:D3:B1:18:CF

Alias name: amzninternalinfoseccag3
Certificate fingerprints:
  MD5:  E9:34:94:02:BA:BB:31:6B:22:E6:2B:A9:C4:F0:26:04
  SHA1: B9:B1:CA:38:F7:BF:9C:D2:D4:95:E7:B6:5E:75:32:9B:A8:78:2E:F6
  SHA256:
 81:03:0B:C7:E2:54:DA:7B:F8:B7:45:DB:DD:41:15:89:B5:A3:81:86:FB:4B:29:77:1F:84:0A:18:D9:67:6D:68

Alias name: cia-crt-g3-02-ca
Certificate fingerprints:
  MD5:  FD:B9:23:FD:D3:EB:2D:3E:57:EF:56:FF:DB:D3:E4:B9
  SHA1: 96:4A:BB:A7:BD:DA:FC:97:34:C0:0A:2D:F0:05:98:F7:E6:C6:6F:09
  SHA256:
 93:F1:72:FB:BA:43:31:5C:06:EE:0F:9F:04:89:B8:F6:88:BC:75:15:3C:BE:B4:80:AC:A7:14:3A:F6:FC:4A:C1

Alias name: entrustrootcertificationauthorityec1
Certificate fingerprints:
  MD5:  B6:7E:1D:F0:58:C5:49:6C:24:3B:3D:ED:98:18:ED:BC
  SHA1: 20:D8:06:40:DF:9B:25:F5:12:25:3A:11:EA:F7:59:8A:EB:14:B5:47
  SHA256:
 02:ED:0E:B2:8C:14:DA:45:16:5C:56:67:91:70:0D:64:51:D7:FB:56:F0:B2:AB:1D:3B:8E:B0:70:E5:6E:DF:F5

Alias name: securitycommunicationrootca
Certificate fingerprints:
  MD5:  F1:BC:63:6A:54:E0:B5:27:F5:CD:E7:1A:E3:4D:6E:4A
  SHA1: 36:B1:2B:49:F9:81:9E:D7:4C:9E:BC:38:0F:C6:56:8F:5D:AC:B2:F7
  SHA256:
 E7:5E:72:ED:9F:56:0E:EC:6E:B4:80:00:73:A4:3F:C3:AD:19:19:5A:39:22:82:01:78:95:97:4A:99:02:6B:6C

Alias name: globalsignca
Certificate fingerprints:
  MD5:  3E:45:52:15:09:51:92:E1:B7:5D:37:9F:B1:87:29:8A
  SHA1: B1:BC:96:8B:D4:F4:9D:62:2A:A8:9A:81:F2:15:01:52:A4:1D:82:9C
  SHA256:
 EB:D4:10:40:E4:BB:3E:C7:42:C9:E3:81:D3:1E:F2:A4:1A:48:B6:68:5C:96:E7:CE:F3:C1:DF:6C:D4:33:1C:99

Alias name: trustcenterclass2caii
Certificate fingerprints:
  MD5:  CE:78:33:5C:59:78:01:6E:18:EA:B9:36:A0:B9:2E:23
  SHA1: AE:50:83:ED:7C:F4:5C:BC:8F:61:C6:21:FE:68:5D:79:42:21:15:6E
```

```
  SHA256:
 E6:B8:F8:76:64:85:F8:07:AE:7F:8D:AC:16:70:46:1F:07:C0:A1:3E:EF:3A:1F:F7:17:53:8D:7A:BA:D3:91:B4

Alias name: camerfirmachambersofcommerceroot
Certificate fingerprints:
  MD5:  B0:01:EE:14:D9:AF:29:18:94:76:8E:F1:69:33:2A:84
  SHA1: 6E:3A:55:A4:19:0C:19:5C:93:84:3C:C0:DB:72:2E:31:30:61:F0:B1
  SHA256:
 0C:25:8A:12:A5:67:4A:EF:25:F2:8B:A7:DC:FA:EC:EE:A3:48:E5:41:E6:F5:CC:4E:E6:3B:71:B3:61:60:6A:C3

Alias name: geotrustprimarycag3
Certificate fingerprints:
  MD5:  B5:E8:34:36:C9:10:44:58:48:70:6D:2E:83:D4:B8:05
  SHA1: 03:9E:ED:B8:0B:E7:A0:3C:69:53:89:3B:20:D2:D9:32:3A:4C:2A:FD
  SHA256:
 B4:78:B8:12:25:0D:F8:78:63:5C:2A:A7:EC:7D:15:5E:AA:62:5E:E8:29:16:E2:CD:29:43:61:88:6C:D1:FB:D4

Alias name: geotrustprimarycag2
Certificate fingerprints:
  MD5:  01:5E:D8:6B:BD:6F:3D:8E:A1:31:F8:12:E0:98:73:6A
  SHA1: 8D:17:84:D5:37:F3:03:7D:EC:70:FE:57:8B:51:9A:99:E6:10:D7:B0
  SHA256:
 5E:DB:7A:C4:3B:82:A0:6A:87:61:E8:D7:BE:49:79:EB:F2:61:1F:7D:D7:9B:F9:1C:1C:6B:56:6A:21:9E:D7:66

Alias name: hongkongpostrootca1
Certificate fingerprints:
  MD5:  A8:0D:6F:39:78:B9:43:6D:77:42:6D:98:5A:CC:23:CA
  SHA1: D6:DA:A8:20:8D:09:D2:15:4D:24:B5:2F:CB:34:6E:B2:58:B2:8A:58
  SHA256:
 F9:E6:7D:33:6C:51:00:2A:C0:54:C6:32:02:2D:66:DD:A2:E7:E3:FF:F1:0A:D0:61:ED:31:D8:BB:B4:10:CF:B2

Alias name: affirmtrustpremiumeccca
Certificate fingerprints:
  MD5:  64:B0:09:55:CF:B1:D5:99:E2:BE:13:AB:A6:5D:EA:4D
  SHA1: B8:23:6B:00:2F:1D:16:86:53:01:55:6C:11:A4:37:CA:EB:FF:C3:BB
  SHA256:
 BD:71:FD:F6:DA:97:E4:CF:62:D1:64:7A:DD:25:81:B0:7D:79:AD:F8:39:7E:B4:EC:BA:9C:5E:84:88:82:14:23

Alias name: hellenicacademicandresearchinstitutionsrootca2015
Certificate fingerprints:
  MD5:  CA:FF:E2:DB:03:D9:CB:4B:E9:0F:AD:84:FD:7B:18:CE
  SHA1: 01:0C:06:95:A6:98:19:14:FF:BF:5F:C6:B0:B6:95:EA:29:E9:12:A6
  SHA256:
 A0:40:92:9A:02:CE:53:B4:AC:F4:F2:FF:C6:98:1C:E4:49:6F:75:5E:6D:45:FE:0B:2A:69:2B:CD:52:52:3F:36
```

IoT Analytics

O IoT Analytics (`iotAnalyticsA` ação envia dados de uma mensagem MQTT para umAWS IoT AnalyticsChannel

Requisitos

Essa ação de regra tem os seguintes requisitos:

- Uma função do IAM queAWS IoTpode presumir para executar `oiotanalytics:BatchPutMessage`operação. Para obter mais informações, consulteConcessão de umAWS IoTgoverne o acesso necessário (p. 479)

  NoAWS IoTconsole, você pode escolher ou criar uma função para permitirAWS IoTPara executar essa ação da regra do.

A política anexada à função especificada deve ser semelhante ao exemplo a seguir.

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

Parâmetros

Ao criar umAWS IoTCom essa ação, você deve especificar as seguintes informações:

`batchMode`

(Opcional) Se a ação deverá ser processada como um lote. O valor padrão é `false`.

Quando`batchMode`é`true`e a instrução SQL de regra é avaliada para uma matriz, cada elemento da matriz é entregue como uma mensagem separada quando transmitido por`BatchPutMessage`para oAWS IoTCanal de análise do. A matriz resultante não pode ter mais de 100 mensagens.

SuporteModelos de substituição (p. 627): Não

`channelName`

O nome do canal do AWS IoT Analytics no qual os dados devem ser gravados.

SuporteModelos de substituição (p. 627): API eAWS CLIapenas

`roleArn`

A função do IAM que permite o acesso aoAWS IoT AnalyticsChannel Para obter mais informações, consulteRequisitos (p. 531)

SuporteModelos de substituição (p. 627): Não

Exemplos

O exemplo JSON a seguir define uma ação do IoT Analytics em umAWS IoTregra.

```
{
    "topicRulePayload": {
        "sql": "SELECT * FROM 'some/topic'",
        "ruleDisabled": false,
        "awsIotSqlVersion": "2016-03-23",
        "actions": [
            {
                "iotAnalytics": {
                    "channelName": "mychannel",
                    "roleArn": "arn:aws:iam::123456789012:role/analyticsRole",
                }
            }
```

```
        ]
    }
}
```

Consulte também

- O que é o AWS IoT Analytics? no Guia do usuário do AWS IoT Analytics
- OAWS IoT Analyticsconsole também tem umInício rápido dorecurso que permite que você crie um canal, armazenamento de dados, pipeline e armazenamento de dados com um clique. Para obter mais informações, consulteAWS IoT AnalyticsGuia de início rápido do consolenoAWS IoT AnalyticsGuia do usuário do.

IoT Events

Os eventos do IoT (`iotEvents`A ação envia dados de uma mensagem MQTT para umAWS IoT Eventsentrada.

Requisitos

Essa ação de regra tem os seguintes requisitos:

- Uma função do IAMAWS IoTpode presumir para executar `oiotevents:BatchPutMessage`operação. Para obter mais informações, consulteConcessão de umAWS IoTgoverne o acesso necessário (p. 479)

  NoAWS IoTconsole, você pode escolher ou criar uma função para permitirAWS IoTPara executar essa ação da regra do.

Parâmetros

Ao criar umAWS IoTCom essa ação, você deve especificar as seguintes informações:

`batchMode`

(Opcional) Se as ações do evento deverão ser processadas como um lote. O valor padrão é `false`.

Quando`batchMode`é`true`e a instrução SQL de regra é avaliada para uma matriz, cada elemento da matriz é tratado como uma mensagem separada quando é enviado paraAWS IoTEventos chamando`BatchPutMessage`. A matriz resultante não pode ter mais de dez mensagens.

Quando `batchMode` é `true`, não é possível especificar um `messageId`.

SuporteModelos de substituição (p. 627): Não

`inputName`

O nome da entrada do AWS IoT Events.

SuporteModelos de substituição (p. 627): API eAWS CLIapenas

`messageId`

(Opcional) Use isso para garantir que apenas uma entrada (mensagem) com determinado`messageId`é processado por umAWS IoT Eventsdetector. Você pode usar o`${newuuid()}`Modelo de substituição para gerar uma ID exclusiva para cada solicitação.

Quando `batchMode` estiver `true`, você não poderá especificar um `messageId`—um novo valor de UUID será atribuído.

SuporteModelos de substituição (p. 627): Sim

`roleArn`

A função do IAM que permiteAWS IoTPara enviar uma entrada para umAWS IoT Eventsdetector. Para obter mais informações, consulteRequisitos (p. 533)

SuporteModelos de substituição (p. 627): Não

Exemplos

O exemplo JSON a seguir define uma ação de IoT Events em umAWS IoTregra.

```
{
    "topicRulePayload": {
        "sql": "SELECT * FROM 'some/topic",
        "ruleDisabled": false,
        "awsIotSqlVersion": "2016-03-23",
        "actions": [
            {
                "iotEvents": {
                    "inputName": "MyIoTEventsInput",
                    "messageId": "${newuuid()}",
                    "roleArn": "arn:aws:iam::123456789012:role/aws_iot_events"
                }
            }
        ]
    }
}
```

Consulte também

- O que é oAWS IoT Events?noAWS IoT EventsGuia do desenvolvedor

IoT SiteWise

A IoT SiteWise (`iotSiteWise`A ação envia dados de uma mensagem MQTT para propriedades de ativo noAWS IoT SiteWise.

É possível seguir um tutorial que mostra como ingerir dados de coisas do AWS IoT. Para obter mais informações, consulteConsumir dados para oAWS IoT SiteWisedesdeAWS IoTcoisasnoAWS IoT SiteWiseGuia do usuário do.

Requisitos

Essa ação de regra tem os seguintes requisitos:

- Uma função do IAMAWS IoTpode presumir para executar `oiotsitewise:BatchPutAssetPropertyValue`operação. Para obter mais informações, consulteConcessão de umAWS IoTgoverne o acesso necessário (p. 479)

  Você pode anexar o exemplo de política de confiança a seguir à função.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "iotsitewise:BatchPutAssetPropertyValue",
            "Resource": "*"
        }
    ]
}
```

  Para melhorar a segurança, você pode especificar um caminho de hierarquia de ativos do AWS IoT SiteWise na propriedade `Condition`. O exemplo a seguir é uma política de confiança que especifica um caminho de hierarquia de ativos.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "iotsitewise:BatchPutAssetPropertyValue",
            "Resource": "*",
            "Condition": {
                "StringLike": {
                    "iotsitewise:assetHierarchyPath": [
                        "/root node asset ID",
                        "/root node asset ID/*"
                    ]
                }
            }
        }
    ]
}
```

- Quando você envia dados do aoAWS IoT SiteWiseCom essa ação, os dados devem atender aos requisitos do`BatchPutAssetPropertyValue`operação. Para obter mais informações, consulte BatchPutAssetPropertyValue na Referência de API do AWS IoT SiteWise.

Parâmetros

Ao criar umAWS IoTCom essa ação, você deve especificar as seguintes informações:

`putAssetPropertyValueEntries`

Uma lista de entradas de valor de propriedade de ativo que contêm as seguintes informações:

`propertyAlias`

(Opcional) O alias da propriedade associado à sua propriedade de ativo. É necessário especificar um `propertyAlias`, ou um `assetId` e um `propertyId`. Para obter mais informações sobre aliases de propriedade, consulte.Mapeamento de fluxos de dados industriais para propriedades de ativosnoAWS IoT SiteWiseGuia do usuário do.

SuporteModelos de substituição (p. 627): Sim

`assetId`

(Opcional) O ID doAWS IoT SiteWiseAtivo. É necessário especificar um `propertyAlias`, ou um `assetId` e um `propertyId`.

SuporteModelos de substituição (p. 627): Sim

`propertyId`

(Opcional) O ID da propriedade do ativo. É necessário especificar um `propertyAlias`, ou um `assetId` e um `propertyId`.

SuporteModelos de substituição (p. 627): Sim

`entryId`

(Opcional) Um identificador exclusivo para essa entrada. Você pode definir o `entryId` para controlar melhor qual mensagem causou um erro em caso de falha. O padrão é um novo UUID.

SuporteModelos de substituição (p. 627): Sim

`propertyValues`

Uma lista de valores de propriedade a serem inseridos, em que cada um contém carimbo de data/hora, qualidade e valor (TQV) no seguinte formato:

`timestamp`

Uma estrutura de carimbo de data/hora que contém as seguintes informações:

`timeInSeconds`

Uma string que contém o tempo em segundos no horário Unix epoch. Se a carga de mensagem não tiver um time stamp, você poderá usar timestamp() (p. 619), que retorna a hora atual em milissegundos. Para converter essa hora em segundos, você pode usar o seguinte modelo de substituição: **`${floor(timestamp() / 1E3)}`**.

SuporteModelos de substituição (p. 627): Sim

`offsetInNanos`

(Opcional) Uma string que contém o deslocamento de tempo nanossegundos do tempo em nanossegundos do tempo em segundos. Se a carga de mensagem não tiver um time stamp, você poderá usar timestamp() (p. 619), que retorna a hora atual em milissegundos. Para calcular o deslocamento em nanossegundos a partir dessa hora, é possível usar o seguinte modelo de substituição: **`${(timestamp() % 1E3) * 1E6}`**.

SuporteModelos de substituição (p. 627): Sim

Com relação ao tempo da época Unix,AWS IoT SiteWiseAceita apenas entradas com um timestamp de até 7 dias no passado e de até 5 minutos no future.

`quality`

(Opcional) Uma string que descreve a qualidade do valor. Valores válidos: `GOOD`, `BAD`, `UNCERTAIN`.

SuporteModelos de substituição (p. 627): Sim

`value`

Uma estrutura de valor que contém um dos seguintes campos de valor, dependendo do tipo de dados da propriedade do ativo:

`booleanValue`

(Opcional) Uma string que contém o valor booleano da entrada de valor.

SuporteModelos de substituição (p. 627): Sim

`doubleValue`

(Opcional) Uma string que contém o valor duplo da entrada de valor.

SuporteModelos de substituição (p. 627): Sim

`integerValue`

(Opcional) Uma string que contém o valor inteiro da entrada de valor.

SuporteModelos de substituição (p. 627): Sim

`stringValue`

(Opcional) O valor da string da entrada de valor.

SuporteModelos de substituição (p. 627): Sim

`roleArn`

O ARN da função do IAM que garanteAWS IoTpermissão para enviar um valor de propriedade de ativo paraAWS IoT SiteWise. Para obter mais informações, consulteRequisitos (p. 535)

SuporteModelos de substituição (p. 627): Não

Exemplos

O exemplo JSON a seguir define uma IoT básica SiteWise ação em umAWS IoTregra.

```
{
    "topicRulePayload": {
        "sql": "SELECT * FROM 'some/topic'",
        "ruleDisabled": false,
        "awsIotSqlVersion": "2016-03-23",
        "actions": [
            {
                "iotSiteWise": {
                    "putAssetPropertyValueEntries": [
                        {
                            "propertyAlias": "/some/property/alias",
                            "propertyValues": [
                                {
                                    "timestamp": {
                                        "timeInSeconds": "${my.payload.timeInSeconds}"
                                    },
                                    "value": {
                                        "integerValue": "${my.payload.value}"
                                    }
                                }
                            ]
                        }
                    ],
                    "roleArn": "arn:aws:iam::123456789012:role/aws_iot_sitewise"
                }
            }
```

```
        ]
    }
}
```

O exemplo JSON a seguir define uma IoT SiteWise ação em umAWS IoTregra. Esse exemplo usa o tópico como o alias da propriedade e a função `timestamp()`. Por exemplo, se você publicar dados em `/company/windfarm/3/turbine/7/rpm`, essa ação enviará os dados para a propriedade do ativo com um alias de propriedade igual ao tópico especificado.

```
{
    "topicRulePayload": {
        "sql": "SELECT * FROM '/company/windfarm/+/turbine/+/+'",
        "ruleDisabled": false,
        "awsIotSqlVersion": "2016-03-23",
        "actions": [
            {
                "iotSiteWise": {
                    "putAssetPropertyValueEntries": [
                        {
                            "propertyAlias": "${topic()}",
                            "propertyValues": [
                                {
                                    "timestamp": {
                                        "timeInSeconds": "${floor(timestamp() / 1E3)}",
                                        "offsetInNanos": "${(timestamp() % 1E3) * 1E6}"
                                    },
                                    "value": {
                                        "doubleValue": "${my.payload.value}"
                                    }
                                }
                            ]
                        }
                    ],
                    "roleArn": "arn:aws:iam::123456789012:role/aws_iot_sitewise"
                }
            }
        ]
    }
}
```

Consulte também

- O que é o AWS IoT SiteWise? no Guia do usuário do AWS IoT SiteWise
- Ingestão de dados usandoAWS IoT Coreregrasno AWS IoT SiteWiseGuia do usuário do
- Consumir dados para oAWS IoT SiteWisedesdeAWS IoTcoisasno AWS IoT SiteWiseGuia do usuário do
- Solução de problemas deAWS IoT SiteWiseAção da regra dono AWS IoT SiteWiseGuia do usuário do

Kinesis Data Firehose

O Kinesis Data Firehose (`firehose`) envia os dados de uma mensagem MQTT para um fluxo do Amazon Kinesis Data Firehose.

Requisitos

Essa ação de regra tem estes requisitos:

- Uma função do IAMAWS IoTpode presumir para executar `firehose:PutRecord`operação. Para obter mais informações, consulteConcessão de umAWS IoTgoverne o acesso necessário (p. 479)

NoAWS IoTconsole, você pode escolher ou criar uma função para permitirAWS IoTPara executar essa ação da regra do.

- Se você usar o Kinesis Data Firehose para enviar dados a um bucket do Amazon S3 e usar umAWS Key Management Service(AWS KMS) Gerenciadas pelo clienteAWS KMS key(chave KMS) para criptografar dados em repouso no Amazon S3, o Kinesis Data Firehose deve ter acesso ao bucket e permissão para usar oAWS KMS keyem nome do chamador. Para obter mais informações, consulteConceda acesso ao Kinesis Data Firehose a um destino do Amazon S3noGuia do desenvolvedor do Amazon Kinesis Data Firehose.

Parâmetros

Ao criar umAWS IoTCom essa ação, você deve especificar as seguintes informações:

`batchMode`

(Opcional) Se o stream do Kinesis Data Firehose deve ser fornecido como um lote usando`PutRecordBatch`. O valor padrão é `false`.

Quando`batchMode`é`true`e a instrução SQL da regra é avaliada para uma matriz, cada elemento da matriz forma um registro no`PutRecordBatch`solicitação. A matriz resultante não pode ter mais de 500 registros.

SuporteModelos de substituição (p. 627): Não

`deliveryStreamName`

O fluxo do Kinesis Data Firehose no qual gravar os dados da mensagem.

SuporteModelos de substituição (p. 627): API eAWS CLIapenas

`separator`

(Opcional) Um separador de caracteres usado para separar registros gravados no fluxo de Kinesis Data Firehose. Se você omitir esse parâmetro, o fluxo não usará separador. Valores válidos:`,`(vírgula),`\t`(guia),`\n`(nova linha),`\r\n`(Nova linha do Windows).

SuporteModelos de substituição (p. 627): Não

`roleArn`

A função do IAM que permite acesso ao fluxo do Kinesis Data Firehose. Para obter mais informações, consulteRequisitos (p. 538)

SuporteModelos de substituição (p. 627): Não

Exemplos

O exemplo JSON a seguir define uma ação do Kinesis Data Firehose em umAWS IoTregra.

```
{
    "topicRulePayload": {
        "sql": "SELECT * FROM 'some/topic'",
        "ruleDisabled": false,
        "awsIotSqlVersion": "2016-03-23",
        "actions": [
            {
                "firehose": {
                    "deliveryStreamName": "my_firehose_stream",
                    "roleArn": "arn:aws:iam::123456789012:role/aws_iot_firehose"
```

```
                }
            }
        ]
    }
}
```

O exemplo JSON a seguir define uma ação do Kinesis Data Firehose com modelos de substituição em umAWS IoTregra.

```
{
    "topicRulePayload": {
        "sql": "SELECT * FROM 'some/topic'",
        "ruleDisabled": false,
        "awsIotSqlVersion": "2016-03-23",
        "actions": [
            {
                "firehose": {
                    "deliveryStreamName": "${topic()}",
                    "roleArn": "arn:aws:iam::123456789012:role/aws_iot_firehose"
                }
            }
        ]
    }
}
```

Consulte também

- O que é o Amazon Kinesis Data Firehose?noGuia do desenvolvedor do Amazon Kinesis Data Firehose

Kinesis Data Streams

O Kinesis Data Streams (`kinesis`) grava dados de uma mensagem MQTT no Amazon Kinesis Data Streams.

Requisitos

Essa ação de regra tem estes requisitos:

- Uma função do IAMAWS IoTpode presumir para executar o`kinesis:PutRecord`operação. Para obter mais informações, consulteConcessão de umAWS IoTgoverne o acesso necessário (p. 479)

  NoAWS IoTconsole, você pode escolher ou criar uma função para permitirAWS IoTPara executar essa ação da regra do.
- Se você usar umAWS Key Management Service(AWS KMS) Gerenciadas pelo clienteAWS KMS key(chave KMS) para criptografar dados em repouso no Kinesis Data Streams, o serviço deve ter permissão para usar aAWS KMS keyem nome do chamador. Para obter mais informações, consultePermissões para usar geradas pelo usuárioAWS KMS keysnoGuia do desenvolvedor do Amazon Kinesis Data Streams.

Parâmetros

Ao criar umAWS IoTCom essa ação, você deve especificar as seguintes informações:

`stream`

O fluxo de dados do Kinesis no qual gravar dados.

SuporteModelos de substituição (p. 627): API eAWS CLIapenas

`partitionKey`

A chave de partição usada para determinar em qual estilhaço os dados são gravados. A chave de partição geralmente é composta por uma expressão (por exemplo,`${topic()}`ou`${timestamp()}`).

SuporteModelos de substituição (p. 627): Sim

`roleArn`

O ARN da função do IAM que garanteAWS IoTConceder permissão para acessar o fluxo de dados do Kinesis. Para obter mais informações, consulteRequisitos (p. 540)

SuporteModelos de substituição (p. 627): Não

Exemplos

O exemplo JSON a seguir define uma ação do Kinesis Data Streams em umAWS IoTregra.

```
{
    "topicRulePayload": {
        "sql": "SELECT * FROM 'some/topic'",
        "ruleDisabled": false,
        "awsIotSqlVersion": "2016-03-23",
        "actions": [
            {
                "kinesis": {
                    "streamName": "my_kinesis_stream",
                    "partitionKey": "${topic()}",
                    "roleArn": "arn:aws:iam::123456789012:role/aws_iot_kinesis"
                }
            }
        ]
    }
}
```

O exemplo JSON a seguir define uma ação do Kinesis com modelos de substituição em umAWS IoTregra.

```
{
    "topicRulePayload": {
        "sql": "SELECT * FROM 'some/topic'",
        "ruleDisabled": false,
        "awsIotSqlVersion": "2016-03-23",
        "actions": [
            {
                "kinesis": {
                    "streamName": "${topic()}",
                    "partitionKey": "${timestamp()}",
                    "roleArn": "arn:aws:iam::123456789012:role/aws_iot_kinesis"
                }
            }
        ]
    }
}
```

Consulte também

- O que é o Amazon Kinesis Data Streams?noGuia do desenvolvedor do Amazon Kinesis Data Streams

Lambda

Um Lambda (`lambda`) ação invoca umAWS Lambda, passando uma mensagem MQTT.AWS IoTO invoca funções do Lambda de forma assíncrona.

É possível seguir um tutorial que mostra como criar e testar uma regra com uma ação do Lambda. Para obter mais informações, consulteTutorial: Formatando uma notificação usando umAWS Lambdafunção (p. 214)

Requisitos

Essa ação de regra tem estes requisitos:

- para oAWS IoTPara invocar uma função do Lambda, é necessário configurar uma política que conceda a função do Lambda`lambda:InvokeFunction`permissão paraAWS IoT. Você só pode invocar uma função do Lambda definida na mesmaRegião da AWSonde sua política do Lambda existe. As funções Lambda usam políticas baseadas em recursos; por isso, você precisa anexar a política à própria função Lambda.

  Use o seguinteAWS CLIComando para anexar uma política que concede ao`lambda:InvokeFunction`permissão.

  ```
  aws lambda add-permission --function-name function_name --region region --principal
   iot.amazonaws.com --source-arn arn:aws:iot:region:account-id:rule/rule_name --source-
  account account-id --statement-id unique_id --action "lambda:InvokeFunction"
  ```

  O`add-permission`O comando espera os seguintes parâmetros:

  `--function-name`

  Nome da função Lambda. Você adiciona uma nova permissão para atualizar a política de recursos da função.

  `--region`

  ORegião da AWSda função do.

  `--principal`

  O diretor que recebe a permissão. Isso deve ser`iot.amazonaws.com`PermitirAWS IoTPermita chamar a função do Lambda.

  `--source-arn`

  O ARN da regra. Você pode usar `oget-topic-rule` AWS CLIPara obter o ARN de uma regra.

  `--source-account`

  OConta da AWSEm que a regra está definida.

  `--statement-id`

  Um identificador de declaração exclusivo.

  `--action`

  A ação do Lambda que você deseja permitir nessa declaração. Para permitirAWS IoTPara chamar uma função do Lambda, especifique`lambda:InvokeFunction`.

  > **Important**
  >
  > Se você adicionar uma permissão para umAWS IoTprincipal sem fornecer o`source-arn`ou`source-account`, qualquerConta da AWSIsso cria uma regra com sua ação do Lambda pode acionar regras para invocar sua função do Lambda da daAWS IoT.

Para obter mais informações, consulte Permissões doAWS Lambda.

- Se você usar umAWS Key Management Service(AWS KMS) Gerenciadas pelo clienteAWS KMS key(Chave do KMS) para criptografar dados em repouso no Lambda, o serviço deve ter permissão para usar aAWS KMS keyem nome do chamador. Para obter mais informações, consulteCriptografia em repousonoAWS LambdaGuia do desenvolvedor.

Parâmetros

Ao criar umAWS IoTCom essa ação, você deve especificar as seguintes informações:

`functionArn`

O ARN da função Lambda a ser invocada.AWS IoTDeve ter permissão para invocar a função. Para obter mais informações, consulteRequisitos (p. 542)

Se você não especificar uma versão ou um alias para sua função Lambda, a versão mais recente da função será executada. Você pode especificar uma versão ou alias se quiser executar uma versão específica da sua função do Lambda. Para especificar uma versão ou alias, anexe a versão ou alias ao ARN da função Lambda.

```
arn:aws:lambda:us-east-2:123456789012:function:myLambdaFunction:someAlias
```

Para obter mais informações sobre versionamento e aliases, consulte.AWS Lambdaversionamento de funções e aliases.

SuporteModelos de substituição (p. 627): API eAWS CLIapenas

Exemplos

O exemplo JSON a seguir define uma ação do Lambda em umAWS IoTregra.

```
{
    "topicRulePayload": {
        "sql": "SELECT * FROM 'some/topic'",
        "ruleDisabled": false,
        "awsIotSqlVersion": "2016-03-23",
        "actions": [
            {
                "lambda": {
                    "functionArn": "arn:aws:lambda:us-
east-2:123456789012:function:myLambdaFunction"
                 }
            }
        ]
    }
}
```

O exemplo JSON a seguir define uma ação do Lambda com modelos de substituição em umAWS IoTregra.

```
{
    "topicRulePayload": {
        "sql": "SELECT * FROM 'some/topic'",
        "ruleDisabled": false,
        "awsIotSqlVersion": "2016-03-23",
```

```
        "actions": [
            {
                "lambda": {
                    "functionArn": "arn:aws:lambda:us-east-1:123456789012:function:
${topic()}"
                }
            }
        ]
    }
}
```

Consulte também

- O que é oAWS Lambda?noAWS LambdaGuia do desenvolvedor
- Tutorial: Formatando uma notificação usando umAWS Lambdafunção (p. 214)

OpenSearch

O OpenSearch (`openSearch`) grava dados de mensagens MQTT em uma Amazon OpenSearch Domínio de serviço. Você pode usar ferramentas como OpenSearch Painéis para consultar e visualizar dados em OpenSearch Serviço.

Requisitos

Essa ação de regra tem estes requisitos:

- Uma função do IAMAWS IoTpode presumir para executar `oes:ESHttpPut`operação. Para obter mais informações, consulteConcessão de umAWS IoTgoverne o acesso necessário (p. 479)

  NoAWS IoTconsole, você pode escolher ou criar uma função para permitirAWS IoTPara executar essa ação da regra do.
- Se você usar um gerenciado pelo clienteAWS KMS key(chave KMS) para criptografar dados em repouso no OpenSearch Serviço, o serviço deve ter permissão para usar a chave do KMS em nome do chamador. Para obter mais informações, consulteCriptografia de dados em repouso para a Amazon OpenSearch ServiçonoAmazônia OpenSearch Guia do desenvolvedor de serviço.

Parâmetros

Ao criar umAWS IoTCom essa ação, você deve especificar as seguintes informações:

`endpoint`

O endpoint da sua Amazon OpenSearch Domínio de serviço.

SuporteModelos de substituição (p. 627): API eAWS CLIapenas

`index`

O OpenSearch O índice do em que você deseja armazenar os dados.

SuporteModelos de substituição (p. 627): Sim

`type`

O tipo de documento que você está armazenando.

SuporteModelos de substituição (p. 627): Sim

`id`

O identificador exclusivo de cada documento.

SuporteModelos de substituição (p. 627): Sim

`roleARN`

A função do IAM que permite o acesso ao OpenSearch Domínio de serviço. Para obter mais informações, consulteRequisitos (p. 544)

SuporteModelos de substituição (p. 627): Não

Limitações

O OpenSearch (`openSearch`) ação não pode ser usada para fornecer dados aos clusters do VPC Elasticsearch.

Exemplos

O exemplo JSON a seguir define um OpenSearch ação em umAWS IoTregra e como você pode especificar os campos para o`openSearch`Ação . Para obter mais informações, consulteOpenSearchAction.

```
{
    "topicRulePayload": {
        "sql": "SELECT *, timestamp() as timestamp FROM 'iot/test'",
        "ruleDisabled": false,
        "awsIotSqlVersion": "2016-03-23",
        "actions": [
            {
                "openSearch": {
                    "endpoint": "https://my-endpoint",
                    "index": "my-index",
                    "type": "my-type",
                    "id": "${newuuid()}",
                    "roleArn": "arn:aws:iam::123456789012:role/aws_iot_os"
                }
            }
        ]
    }
}
```

O exemplo JSON a seguir define um OpenSearch ação com modelos de substituição em umAWS IoTregra.

```
{
    "topicRulePayload": {
        "sql": "SELECT * FROM 'some/topic'",
        "ruleDisabled": false,
        "awsIotSqlVersion": "2016-03-23",
        "actions": [
            {
                "openSearch": {
                    "endpoint": "https://my-endpoint",
                    "index": "${topic()}",
                    "type": "${type}",
                    "id": "${newuuid()}",
                    "roleArn": "arn:aws:iam::123456789012:role/aws_iot_os"
                }
            }
```

```
        ]
    }
}
```

Consulte também

O que é o Amazon OpenSearch Serviço?noAmazônia OpenSearch Guia do desenvolvedor do serviço

Republish

A republicação (`republish`) A ação republica uma mensagem MQTT em outro tópico MQTT.

Requisitos

Essa ação de regra tem estes requisitos:

- Uma função do IAMAWS IoTpode presumir para executar `oiot:Publish`operação. Para obter mais informações, consulteConcessão de umAWS IoTgoverne o acesso necessário (p. 479)

  NoAWS IoTconsole, você pode escolher ou criar uma função para permitirAWS IoTPara executar essa ação da regra do.

Parâmetros

Ao criar umAWS IoTCom essa ação, você deve especificar as seguintes informações:

`topic`

O tópico MQTT no qual publicar a mensagem novamente.

Para republicar em um tópico reservado, que começa com`$`, usar`$$`Em vez disso. Por exemplo, para republicar no tópico de sombra do dispositivo`$aws/things/MyThing/shadow/update`, especifique o tópico como`$$aws/things/MyThing/shadow/update`.

> **Note**
>
> Publicar novamente paraTópicos de trabalhos reserv (p. 109)não é compatível.

SuporteModelos de substituição (p. 627): Sim

`qos`

(Opcional) O nível Quality of Service (QoS) a ser usado ao republicar mensagens. Valores válidos: `0`,`1`. O valor padrão é`0`. Para obter mais informações sobre o QoS do MQTT, consulteMQTT (p. 86).

SuporteModelos de substituição (p. 627): Não

`roleArn`

A função do IAM que permiteAWS IoTPara publicar no tópico MQTT. Para obter mais informações, consulteRequisitos (p. 546)

SuporteModelos de substituição (p. 627): Não

Exemplos

O exemplo JSON a seguir define uma ação de republicação em umAWS IoTregra.

```
{
```

```
    "topicRulePayload": {
        "sql": "SELECT * FROM 'some/topic'",
        "ruleDisabled": false,
        "awsIotSqlVersion": "2016-03-23",
        "actions": [
            {
                "republish": {
                    "topic": "another/topic",
                    "qos": 1,
                    "roleArn": "arn:aws:iam::123456789012:role/aws_iot_republish"
                }
            }
        ]
    }
}
```

O exemplo JSON a seguir define uma ação de republicação com modelos de substituição em umAWS IoTregra.

```
{
    "topicRulePayload": {
        "sql": "SELECT * FROM 'some/topic'",
        "ruleDisabled": false,
        "awsIotSqlVersion": "2016-03-23",
        "actions": [
            {
                "republish": {
                    "topic": "${topic()}/republish",
                    "roleArn": "arn:aws:iam::123456789012:role/aws_iot_republish"
                }
            }
        ]
    }
}
```

S3

O S3 (`s3`) grava os dados de uma mensagem MQTT em um bucket do Amazon Simple Storage Service (Amazon S3).

Requisitos

Essa ação de regra tem estes requisitos:

- Uma função do IAMAWS IoTpode presumir para executar `os3:PutObject`operação. Para obter mais informações, consulteConcessão de umAWS IoTgoverne o acesso necessário (p. 479)

  NoAWS IoTconsole, você pode escolher ou criar uma função para permitirAWS IoTPara executar essa ação da regra do.
- Se você usar umAWS Key Management Service(AWS KMS) Gerenciadas pelo clienteAWS KMS key(chave KMS) para criptografar dados em repouso no Amazon S3, o serviço deve ter permissão para usar aAWS KMS keyem nome do chamador. Para obter mais informações, consulteAWSgerenciadoAWS KMS keysgerenciada pelo clienteAWS KMS keysnoGuia do desenvolvedor do Amazon Simple Storage Service.

Parâmetros

Ao criar umAWS IoTCom essa ação, você deve especificar as seguintes informações:

`bucket`

O bucket do Amazon S3 no qual gravar dados.

SuporteModelos de substituição (p. 627): API eAWS CLIapenas

`cannedacl`

(Opcional) A ACL pré-configurada do Amazon S3 que controla o acesso ao objeto identificado pela chave de objeto. Para obter mais informações, incluindo valores permitidos, consulte.ACL pré-configurada.

SuporteModelos de substituição (p. 627): Não

`key`

O caminho para o arquivo em que os dados são gravados.

Considere um exemplo em que esse parâmetro é`${topic()}/${timestamp()}`e a regra recebe uma mensagem em que o tópico está`some/topic`. Se o carimbo de data/hora atual for`1460685389`, então esta ação grava os dados em um arquivo chamado`1460685389`no`some/topic`pasta do bucket do S3.

> **Note**
>
> Se você usar uma chave estática,AWS IoTsubstitui um único arquivo toda vez que a regra for chamada. Recomendamos que você use o timestamp da mensagem ou outro identificador exclusivo da mensagem para que um novo arquivo seja salvo no Amazon S3 para cada mensagem recebida.

SuporteModelos de substituição (p. 627): Sim

`roleArn`

A função do IAM que permite o acesso ao bucket do Amazon S3. Para obter mais informações, consulteRequisitos (p. 547)

SuporteModelos de substituição (p. 627): Não

Exemplos

O exemplo JSON a seguir define uma ação do S3 em umAWS IoTregra.

```
{
    "topicRulePayload": {
        "sql": "SELECT * FROM 'some/topic'",
        "ruleDisabled": false,
        "awsIotSqlVersion": "2016-03-23",
        "actions": [
            {
                "s3": {
                    "bucketName": "my-bucket",
                    "cannedacl": "public-read",
                    "key": "${topic()}/${timestamp()}",
                    "roleArn": "arn:aws:iam::123456789012:role/aws_iot_s3"
                }
            }
        ]
    }
}
```

Consulte também

- O que é o Amazon S3?noGuia do usuário do Amazon Simple Storage Service

Salesforce IoT

O Salesforce IoT (`salesforce`) envia os dados da mensagem MQTT que acionou a regra para um fluxo de entrada da Salesforce IoT.

Parâmetros

Ao criar umAWS IoTCom essa ação, você deve especificar as seguintes informações:

`url`

O URL exposto pelo stream de entrada da Salesforce IoT. O URL está disponível na plataforma da Salesforce IoT ao criar um stream de entrada. Para obter mais informações, consulte a documentação da Salesforce IoT.

SuporteModelos de substituição (p. 627): Não

`token`

O token usado para autenticar o acesso ao stream de entrada da Salesforce IoT especificado. O token está disponível na plataforma da Salesforce IoT ao criar um stream de entrada. Para obter mais informações, consulte a documentação da Salesforce IoT.

SuporteModelos de substituição (p. 627): Não

Exemplos

O exemplo JSON a seguir define uma ação do Salesforce IoT em umAWS IoTregra.

```
{
    "topicRulePayload": {
        "sql": "SELECT * FROM 'some/topic'",
        "ruleDisabled": false,
        "awsIotSqlVersion": "2016-03-23",
        "actions": [
            {
                "salesforce": {
                    "token": "ABCDEFGHI123456789abcdefghi123456789",
                    "url": "https://ingestion-cluster-id.my-env.sfdcnow.com/streams/stream-
id/connection-id/my-event"
                }
            }
        ]
    }
}
```

SNS

O SNS (`sns`) envia os dados de uma mensagem MQTT como uma notificação por push do Amazon Simple Notification Service (Amazon SNS).

É possível seguir um tutorial que mostra como criar e testar uma regra com uma ação do SNS. Para obter mais informações, consulteTutorial: Enviando notificação do Amazon SNS (p. 201)

Note

A ação SNS não é compatível comTópicos FIFO do Amazon SNS (First-Out). Como o mecanismo de regras é um serviço totalmente distribuído, não há garantia da ordem das mensagens quando a ação do SNS é invocada.

Requisitos

Essa ação de regra tem estes requisitos:

- Uma função do IAMAWS IoTpode presumir para executar `osns:Publish`operação. Para obter mais informações, consulteConcessão de umAWS IoTgoverne o acesso necessário (p. 479)

  NoAWS IoTconsole, você pode escolher ou criar uma função para permitirAWS IoTPara executar essa ação da regra do.
- Se você usar umAWS Key Management Service(AWS KMS) Gerenciadas pelo clienteAWS KMS key(chave KMS) para criptografar dados em repouso no Amazon SNS, o serviço deve ter permissão para usar aAWS KMS keyem nome do chamador. Para obter mais informações, consulteGerenciamento de chavesnoAmazon Simple Notification Service.

Parâmetros

Ao criar umAWS IoTCom essa ação, você deve especificar as seguintes informações:

`targetArn`

O tópico do SNS ou o dispositivo individual para o qual a notificação por push é enviada.

SuporteModelos de substituição (p. 627): API eAWS CLIapenas

`messageFormat`

(Opcional) O formato da mensagem. O Amazon SNS usa essa configuração para determinar se a carga útil deve ser analisada e se partes relevantes específicas da plataforma da carga útil devem ser extraídas. Valores válidos: `JSON`,`RAW`. O padrão é`RAW`.

SuporteModelos de substituição (p. 627): Não

`roleArn`

A função do IAM que permite o acesso ao SNS. Para obter mais informações, consulteRequisitos (p. 550)

SuporteModelos de substituição (p. 627): Não

Exemplos

O exemplo JSON a seguir define uma ação do SNS em umAWS IoTregra.

```
{
    "topicRulePayload": {
        "sql": "SELECT * FROM 'some/topic'",
        "ruleDisabled": false,
        "awsIotSqlVersion": "2016-03-23",
        "actions": [
            {
                "sns": {
```

```
                    "targetArn": "arn:aws:sns:us-east-2:123456789012:my_sns_topic",
                    "roleArn": "arn:aws:iam::123456789012:role/aws_iot_sns"
                }
            }
        ]
    }
}
```

O exemplo JSON a seguir define uma ação do SNS com modelos de substituição em umAWS IoTregra.

```
{
    "topicRulePayload": {
        "sql": "SELECT * FROM 'some/topic'",
        "ruleDisabled": false,
        "awsIotSqlVersion": "2016-03-23",
        "actions": [
            {
                "sns": {
                    "targetArn": "arn:aws:sns:us-east-1:123456789012:${topic()}",
                    "messageFormat": "JSON",
                    "roleArn": "arn:aws:iam::123456789012:role/aws_iot_sns"
                }
            }
        ]
    }
}
```

Consulte também

- O que é o Amazon Simple Notification Service?noAmazon Simple Notification Service
- Tutorial: Enviando notificação do Amazon SNS (p. 201)

SQS

O SQS (`sqs`) da ação da envia os dados de uma mensagem MQTT para uma fila do Amazon Simple Queue Service (Amazon SQS).

> **Note**
>
> A ação SQS não é compatível comFilas FIFO (primeiro a entrar, primeiro a sair) do Amazon SQS. Como o mecanismo de regras é um serviço totalmente distribuído, não há garantia da ordem das mensagens quando a ação do SQS é acionada.

Requisitos

Essa ação de regra tem estes requisitos:

- Uma função do IAMAWS IoTpode presumir para executar o`sqs:SendMessage`operação. Para obter mais informações, consulteConcessão de umAWS IoTgoverne o acesso necessário (p. 479)

  NoAWS IoTconsole, você pode escolher ou criar uma função para permitirAWS IoTPara executar essa ação da regra do.
- Se você usar umAWS Key Management Service(AWS KMS) Gerenciadas pelo clienteAWS KMS key(chave KMS) para criptografar dados em repouso no Amazon SQS, o serviço deve ter permissão para usar aAWS KMS keyem nome do chamador. Para obter mais informações, consulteGerenciamento de chavesnoGuia do desenvolvedor do Amazon Simple Queue Service.

Parâmetros

Ao criar umAWS IoTCom essa ação, você deve especificar as seguintes informações:

`queueUrl`

O URL da fila do Amazon SQS na qual gravar os dados.

SuporteModelos de substituição (p. 627): API eAWS CLIapenas

`useBase64`

Defina este parâmetro como`true`para configurar a ação de regra para codificar os dados da mensagem base64 antes de gravar os dados na fila do Amazon SQS. Padronizado como `false`.

SuporteModelos de substituição (p. 627): Não

`roleArn`

A função do IAM que permite o acesso à fila do Amazon SQS. Para obter mais informações, consulteRequisitos (p. 551)

SuporteModelos de substituição (p. 627): Não

Exemplos

O exemplo JSON a seguir define uma ação SQS em umAWS IoTregra.

```
{
    "topicRulePayload": {
        "sql": "SELECT * FROM 'some/topic'",
        "ruleDisabled": false,
        "awsIotSqlVersion": "2016-03-23",
        "actions": [
            {
                "sqs": {
                    "queueUrl": "https://sqs.us-east-2.amazonaws.com/123456789012/
my_sqs_queue",
                    "roleArn": "arn:aws:iam::123456789012:role/aws_iot_sqs"
                }
            }
        ]
    }
}
```

O exemplo JSON a seguir define uma ação SQS com modelos de substituição em umAWS IoTregra.

```
{
    "topicRulePayload": {
        "sql": "SELECT * FROM 'some/topic'",
        "ruleDisabled": false,
        "awsIotSqlVersion": "2016-03-23",
        "actions": [
            {
                "sqs": {
                    "queueUrl": "https://sqs.us-east-2.amazonaws.com/123456789012/
${topic()}",
                    "useBase64": true,
                    "roleArn": "arn:aws:iam::123456789012:role/aws_iot_sqs"
                }
```

```
            }
        ]
    }
}
```

Consulte também

- O que é Amazon Simple Queue Service? no Guia do desenvolvedor do Amazon Simple Queue Service

Step Functions

Functions do Step (`stepFunctions`) a ação inicia umAWS Step Functionsmáquina de estado do.

Requisitos

Essa ação de regra tem estes requisitos:

- Uma função do IAMAWS IoTpode presumir para executar o`states:StartExecution`operação. Para obter mais informações, consulteConcessão de umAWS IoTgoverne o acesso necessário (p. 479)

  NoAWS IoTconsole, você pode escolher ou criar uma função para permitirAWS IoTPara executar essa ação da regra do.

Parâmetros

Ao criar umAWS IoTCom essa ação, você deve especificar as seguintes informações:

`stateMachineName`

O nome da máquina de estados do Step Functions a ser iniciada.

SuporteModelos de substituição (p. 627): API eAWS CLIapenas

`executionNamePrefix`

(Opcional) O nome atribuído à execução da máquina de estado consiste desse prefixo seguido por um UUID. O Step Functions cria um nome exclusivo para cada execução da máquina de estados, caso um nome não seja fornecido.

SuporteModelos de substituição (p. 627): Sim

`roleArn`

O ARN da função do que garanteAWS IoTPermita iniciar a máquina de estado. Para obter mais informações, consulteRequisitos (p. 553)

SuporteModelos de substituição (p. 627): Não

Exemplos

O exemplo JSON a seguir define uma ação Step Functions em umAWS IoTregra.

```
{
    "topicRulePayload": {
```

```
        "sql": "SELECT * FROM 'some/topic'",
        "ruleDisabled": false,
        "awsIotSqlVersion": "2016-03-23",
        "actions": [
            {
                "stepFunctions": {
                    "stateMachineName": "myStateMachine",
                    "executionNamePrefix": "myExecution",
                    "roleArn": "arn:aws:iam::123456789012:role/aws_iot_step_functions"
                }
            }
        ]
    }
}
```

Consulte também

- O que é oAWS Step Functions?noAWS Step FunctionsGuia do desenvolvedor

Timestream

A ação da regra Timestream grava atributos (medidas) de uma mensagem MQTT em uma tabela do Amazon Timestream. Para obter mais informações sobre o Amazon Timestream, consulte.O que é o Amazon Timestream?.

> **Note**
>
> O Amazon Timestream não está disponível em todas asRegião da AWSs. Se o Amazon Timestream não estiver disponível em sua região, ele não aparecerá na lista de ações de regras.

Os atributos que essa regra armazena no banco de dados Timestream são aqueles que resultam da instrução de consulta da regra. O valor de cada atributo no resultado da instrução de consulta é analisado para inferir seu tipo de dados (como em umthe section called “DynamoDBv2” (p. 501)ação). O valor de cada atributo é gravado em seu próprio registro na tabela Timestream. Para especificar ou alterar o tipo de dados de um atributo, use a`cast()` (p. 587)na instrução da consulta. Para obter mais informações sobre o conteúdo de cada registro do Timestream, consultethe section called “Conteúdo do registro do Amazon Timestream” (p. 556).

> **Note**
>
> Com SQL V2 (2016-03-23), valores numéricos que são números inteiros, como`10.0`, são convertidos sua representação Integer (`10`). Lançando-os explicitamente para um`Decimal`valor, como usando ocast() (p. 587)função, não impede esse comportamento — o resultado ainda é um`Integer`value. Isso pode causar erros de incompatibilidade de tipo que impedem que dados sejam registrados no banco de dados do Timestream. Para processar com segurança valores numéricos de número inteiro como`Decimal`, use SQL V1 (2015-10-08) para a instrução de consulta de regra.

> **Note**
>
> O número máximo de valores que uma ação de regra Timestream pode gravar em uma tabela do Amazon Timestream é 100. Para obter mais informações, consulteReferência da cota Amazon Timestream.

Requisitos

Essa ação de regra tem estes requisitos:

- Uma função do IAMAWS IoTpode presumir para executar `otimestream:DescribeEndpointsetimestream:WriteRecords`operações. Para obter mais informações, consulteConcessão de umAWS IoTgoverne o acesso necessário (p. 479)

  NoAWS IoTconsole, você pode escolher, atualizar ou criar uma função para permitirAWS IoTPara executar essa ação da regra do.
- Se você usar um gerenciado pelo clienteAWS Key Management Service(AWS KMS) Para criptografar dados em repouso no Timestream, o serviço deve ter permissão para usar aAWS KMS keyem nome do chamador. Para obter mais informações, consulteComoAWSUso de serviçosAWSKMS.

Parâmetros

Ao criar umAWS IoTCom essa ação, você deve especificar as seguintes informações:

`databaseName`

O nome de um banco de dados do Amazon Timestream que tem a tabela para receber os registros que essa ação cria. Consulte também **`tableName`**.

SuporteModelos de substituição (p. 627): API eAWS CLIapenas

`dimensions`

Atributos de metadados das séries temporais que são gravados em cada registro de medida. Por exemplo, o nome e a zona de disponibilidade de uma instância do EC2 ou o nome do fabricante de uma turbina eólica são dimensões.

`name`

O nome da dimensão de metadados. Este é o nome da coluna no registro da tabela do banco de dados.

As dimensões não podem ser nomeadas:`measure_name`,`measure_value`, ou`time`. Esses nomes são reservados. Nomes de dimensões não podem começar com`ts_`ou`measure_value`e eles não podem conter o cólon (`:`) caractere.

SuporteModelos de substituição (p. 627): Não

`value`

O valor a ser gravado nesta coluna do registro do banco de dados.

SuporteModelos de substituição (p. 627): Sim

`roleArn`

O nome de recurso da Amazon (ARN) da função que concede à função do que concedeAWS IoTpermissão para gravar na tabela de banco de dados Timestream. Para obter mais informações, consulteRequisitos (p. 554)

SuporteModelos de substituição (p. 627): Não

`tableName`

O nome da tabela do banco de dados na qual os registros de medida devem ser gravados. Consulte também **`databaseName`**.

SuporteModelos de substituição (p. 627): API eAWS CLIapenas

`timestamp`

O valor a ser usado para o timestamp da entrada. Se estiver em branco, a hora em que a entrada foi processada será usada.

`unit`

A precisão do valor do timestamp que resulta da expressão descrita em `value`.

Valores válidos: `SECONDS` |`MILLISECONDS`|`MICROSECONDS`|`NANOSECONDS`. O padrão é`MILLISECONDS`.

`value`

Uma expressão que retorna um valor de tempo epoch longo.

Você pode usar othe section called "time_to_epoch (String, String)" (p. 618)para criar um carimbo de data/hora válido a partir de um valor de data ou hora passado na carga da mensagem.

Conteúdo do registro do Amazon Timestream

Os dados gravados na tabela do Amazon Timestream por essa ação incluem um carimbo de data/hora, metadados da ação de regra Timestream e o resultado da instrução de consulta da regra.

Para cada atributo (medida) no resultado da instrução de consulta, essa ação de regra grava um registro na tabela Timestream especificada com essas colunas.

Nome da coluna	Tipo de atributo	Valor	Comentários
`nome da dimensão`	DIMENSÃO	O valor especificado na entrada de ação da regra Timestream.	Cada umDimensãoespecificado na entrada de ação de regra cria uma coluna no banco de dados Timestream com o nome da dimensão.
measure name	MEASURE NAME	O nome do atributo	O nome do atributo no resultado da instrução de consulta cujo valor é especificado no`measure_value::`*`data-type`*coluna.
measure _value#*`Tipo de dados`*	MEASURE _VALUE	O valor do atributo no resultado da instrução de consulta. O nome do atributo está no`measure_name`coluna.	O valor é interpretado* e lançado como a melhor combinação de:`bigint`,`boolean`,`double`, ou`varchar`. O Amazon Timestream cria uma coluna separada para cada tipo de dados. O valor na mensagem pode ser convertido para outro tipo de dados usando o`cast()` (p. 587)função na instrução de consulta da regra.
time	TIMESTAMP	A data e a hora do registro no banco de dados.	Esse valor é atribuído pelo mecanismo de regras ou

Nome da coluna	Tipo de atributo	Valor	Comentários
			pelo`timestamp`propriedade, se estiver definida.

* O valor do atributo lido da carga útil da mensagem é interpretado da seguinte forma. Consulte othe section called “Exemplos” (p. 557)Para obter uma ilustração de cada um desses casos.

- Um valor não citado de`true`ou`false`é interpretado como um`boolean`type.
- Um numérico decimal é interpretado como um`double`type.
- Um valor numérico sem um ponto decimal é interpretado como um`bigint`type.
- Uma string entre aspas é interpretada como uma`varchar`type.
- Objetos e valores de matriz são convertidos em strings JSON e armazenados como um`varchar`type.

Exemplos

O exemplo JSON a seguir define uma ação de regra Timestream com um modelo de substituição em umAWS IoTregra.

```
{
  "topicRulePayload": {
    "sql": "SELECT * FROM 'iot/topic'",
    "ruleDisabled": false,
    "awsIotSqlVersion": "2016-03-23",
    "actions": [
      {
        "timestream": {
          "roleArn": "arn:aws:iam::123456789012:role/aws_iot_timestream",
          "tableName": "devices_metrics",
          "dimensions": [
            {
              "name": "device_id",
              "value": "${clientId()}"
            },
            {
              "name": "device_firmware_sku",
              "value": "My Static Metadata"
            }
          ],
          "databaseName": "record_devices"
        }
      }
    ]
  }
}
```

Usar a ação de regra de tópico Timestream definida no exemplo anterior com a seguinte carga de mensagem resulta nos registros do Amazon Timestream gravados na tabela a seguir.

```
{
  "boolean_value": true,
  "integer_value": 123456789012,
  "double_value": 123.456789012,
  "string_value": "String value",
  "boolean_value_as_string": "true",
  "integer_value_as_string": "123456789012",
  "double_value_as_string": "123.456789012",
  "array_of_integers": [23,36,56,72],
```

```
  "array of strings": ["red", "green","blue"],
  "complex_value": {
    "simple_element": 42,
    "array_of_integers": [23,36,56,72],
    "array of strings": ["red", "green","blue"]
  }
}
```

A tabela a seguir exibe as colunas e registros do banco de dados que usam a ação de regra de tópico especificada para processar a mensagem anterior que o payload cria. O`device_firmware_sku`e`device_id`colunas são as DIMENSÕES definidas na ação de regra de tópico. A ação da regra de tópico Timestream cria o`time`coluna e o`measure_name`e`measure_value::*`colunas, que ele preenche com os valores do resultado da instrução de consulta da ação de regra de tópico.

device_firmw	device_id	measure name	measure _value# bigint	measure _value# varchar	measure _value# double	measure _value# booleano	time
Meus metadados estáticos	IOTconsole-1594exemplo738-0	complex_value	-	{"simple_element” :42, "array_of_integers”: [23,36,56,72], "array de strings”: ["vermelho”, "verde”, "azul "]}	-	-	24/08/2020
Meus metadados estáticos	IOTconsole-1594exemplo738-0	integer_pouco_as_string	-	123456789012	-	-	24/08/2020
Meus metadados estáticos	IOTconsole-1594exemplo738-0	boolean_value	-	-	-	TRUE	24/08/2020
Meus metadados estáticos	IOTconsole-1594exemplo738-0	integer_pouco	123456789012	-	-	-	24/08/2020
Meus metadados estáticos	IOTconsole-1594exemplo738-0	string_valor	-	Valor da string	-	-	24/08/2020
Meus metadados estáticos	IOTconsole-1594exemplo738-0	array_of_integers	-	[23.36,56,72]	-	-	24/08/2020
Meus metadados estáticos	IOTconsole-1594exemplo738-0	array de strings	-	["vermelho”, "verde”, "azul "]	-	-	24/08/2020
Meus metadados estáticos	IOTconsole-1594exemplo738-0	boolean_pouco_as_string	-	TRUE	-	-	24/08/2020
Meus metadados estáticos	IOTconsole-1594exemplo738-0	double_pouco	-	-	123.456789012	-	24/08/2020

device_firmw	device_id	measure name	measure _value# bigint	measure _value# varchar	measure _value# double	measure _value# booleano	time
Meus metadados estáticos	IOTconsole-1[illegible]	[illegible]	[illegible]	123.45679	-	-	24/08/2020 22:42:16 .423000000

Solucionar problemas de uma regra

Se tiver algum problema com as regras, habilite CloudWatch Logs. É possível analisar seus logs para determinar se o problema é autorização ou se, por exemplo, uma condição de cláusula WHERE não foi correspondida. Para obter mais informações, consulteConfigurar o CloudWatch Logs.

Como acessar recursos entre contas usandoAWS IoTregras

Você pode configurarAWS IoTregras para acesso entre contas para que os dados ingeridos em tópicos MQTT de uma conta possam ser roteados para oAWSserviços, como Amazon SQS e Lambda, de outra conta. O seguinte explica como configurar oAWS IoTregras para ingestão de dados entre contas, de um tópico MQTT em uma conta, para um destino em outra conta.

Regras entre contas podem ser configuradas usandoPermissões baseadas em recursosno recurso de destino. Portanto, somente destinos que suportam permissões baseadas em recursos podem ser ativados para o acesso entre contas comAWS IoTregras. Os destinos com suporte incluem Amazon SQS, Amazon SNS, Amazon S3 eAWS Lambda.

> **Note**
>
> Você deve definir a regra na mesmaRegião da AWScomo recurso de outro serviço, para que a ação da regra possa interagir com esse recurso. Para obter mais informações sobreAWS IoTAções de regras, consulteAWS IoTAções de regra do (p. 485).

Pré-requisitos

- Familiaridade com oAWS IoTregras
- Um entendimento do IAMusuários,papéis, ePermissão baseada em recursos
- TerAWS CLIinstalada

Configuração entre contas para o Amazon SQS

Cenário: A Conta A envia os dados de uma mensagem MQTT para a fila do Amazon SQS da conta B.

Conta da AWS	Conta referida como	Descrição
`1111-1111-1111`	Conta A	Ação da regra:`sqs:SendMessage`

Conta da AWS	Conta referida como	Descrição
2222-2222-2222	Conta B	Fila do Amazon SQS • Nome de região da Amazon (ARN): *arn:aws:sqs:region:2222-2222-2222:ExampleQueue* • URL:*https://sqs.region.amazonaws.com/2222-2222-2222/ExampleQueue*

Faça as tarefas da Conta A

Observação

Para executar os seguintes comandos, o usuário do IAM deve ter permissões para`iot:CreateTopicRule`Com o nome de recurso da Amazon (ARN) da regra como recurso e permissões para`iam:PassRole`ação com um recurso como o ARN da função.

1. ConfigureAWS CLIusando o usuário IAM da conta A.
2. Crie uma função do IAM que confiaAWS IoTmecanismo de regras e anexa uma política que permite o acesso à fila Amazon SQS da conta B. Veja exemplos de comandos e documentos de política emConcessãoAWS IoTo acesso necessário ao.
3. Para criar uma regra anexada a um tópico, execute ocomando create-topic-rule.

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

Veja a seguir um exemplo de arquivo de carga útil com uma regra que insere todas as mensagens enviadas para `aiot/test`tópico na fila especificada do Amazon SQS. A declaração do SQL filtra as mensagens, e o ARN da função concedeAWS IoTPermissões para adicionar a mensagem à fila do Amazon SQS.

```
{
 "sql": "SELECT * FROM 'iot/test'",
 "ruleDisabled": false,
 "awsIotSqlVersion": "2016-03-23",
 "actions": [
  {
   "sqs": {
    "queueUrl": "https://sqs.region.amazonaws.com/2222-2222-2222/ExampleQueue",
    "roleArn": "arn:aws:iam::1111-1111-1111:role/my-iot-role",
    "useBase64": false
   }
  }
 ]
}
```

Para obter mais informações sobre como definir uma ação do Amazon SQS em umaAWS IoTregra, consulteAWS IoTações de regra - Amazon SQS.

Faça as tarefas da Conta B

1. ConfigureAWS CLIusando o usuário IAM da conta B.
2. Para conceder permissões para o recurso de fila do Amazon SQS para a conta A, execute ocomando add-permission.

```
aws sqs add-permission --queue-url https://sqs.region.amazonaws.com/2222-2222-2222/
ExampleQueue --label SendMessagesToMyQueue --aws-account-ids 1111-1111-1111 --actions
 SendMessage
```

Configuração entre contas do Amazon SNS

Cenário: A Conta A envia os dados de uma mensagem MQTT para um tópico do Amazon SNS da conta B.

Conta da AWS	Conta referida como	Descrição
1111-1111-1111	Conta A	Ação da regra:`sns:Publish`
2222-2222-2222	Conta B	ARN do tópico do Amazon SNS:*arn:aws:sns:region:2222-2222-2222:ExampleTopic*

Faça as tarefas da Conta A

Observações

Para executar os seguintes comandos, o usuário do IAM deve ter permissões para`iot:CreateTopicRule`com o ARN de regra como recurso e permissões para `oiam:PassRole`ação com um recurso como função ARN.

1. ConfigureAWS CLIusando o usuário IAM da conta A.
2. Crie uma função do IAM que confiaAWS IoTmecanismo de regras e anexa uma política que permite acesso ao tópico Amazon SNS da conta B. Por exemplo, comandos e documentos de política, consulteConcessãoAWS IoTo acesso necessário ao.
3. Para criar uma regra anexada a um tópico, execute ocomando create-topic-rule.

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

Veja a seguir um exemplo de arquivo de carga útil com uma regra que insere todas as mensagens enviadas para `aiot/test`tópico no tópico especificado do Amazon SNS. A declaração do SQL filtra as mensagens, e a função concede ao ARN da função concedeAWS IoTPermissões para enviar a mensagem para o tópico do Amazon SNS.

```
{
 "sql": "SELECT * FROM 'iot/test'",
 "ruleDisabled": false,
 "awsIotSqlVersion": "2016-03-23",
 "actions": [
  {
   "sns": {
    "targetArn": "arn:aws:sns:region:2222-2222-2222:ExampleTopic",
    "roleArn": "arn:aws:iam::1111-1111-1111:role/my-iot-role"
   }
  }
 ]
}
```

Para obter mais informações sobre como definir uma ação do Amazon SNS em umaAWS IoTregra, consulteAWS IoTações de regras — Amazon SNS.

Faça as tarefas da Conta B

1. ConfigureAWS CLIusando o usuário IAM da conta B.
2. Para conceder permissão no recurso de tópico do Amazon SNS para a conta A, execute ocomando add-permission.

```
aws sns add-permission --topic-arn arn:aws:sns:region:2222-2222-2222:ExampleTopic --
label Publish-Permission --aws-account-id 1111-1111-1111 --action-name Publish
```

Configuração entre contas do Amazon S3

Cenário: A Conta A envia os dados de uma mensagem MQTT para um bucket da conta do Amazon S3 B.

Conta da AWS	Conta referida como	Descrição
1111-1111-1111	Conta A	Ação da regra:`s3:PutObject`
2222-2222-2222	Conta B	ARN do bucket do Amazon S3:*arn:aws:s3:::ExampleBucket*

Faça as tarefas da Conta A

Observação

Para executar os seguintes comandos, o usuário do IAM deve ter permissões para`iot:CreateTopicRule`com a regra ARN como um recurso e permissões para`iam:PassRole`ação com um recurso como função ARN.

1. ConfigureAWS CLIusando o usuário IAM da conta A.
2. Crie uma função do IAM que confiaAWS IoTO mecanismo de regras e anexa uma política que permite acesso ao bucket do Amazon S3 da conta B. Por exemplo, comandos e documentos de política, consulteConcessãoAWS IoTo acesso necessário ao.
3. Para criar uma regra anexada ao bucket do S3 de destino, execute ocomando create-topic-rule.

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

Veja a seguir um exemplo de arquivo de carga útil com uma regra que insere todas as mensagens enviadas para `aiot/test`tópico no bucket do Amazon S3 especificado. A declaração do SQL filtra as mensagens, e a função concede ao ARN da função concedeAWS IoTPermissões para adicionar a mensagem ao bucket do Amazon S3.

```
{
 "sql": "SELECT * FROM 'iot/test'",
 "ruleDisabled": false,
 "awsIotSqlVersion": "2016-03-23",
 "actions": [
  {
   "s3": {
    "bucketName": "ExampleBucket",
    "key": "${topic()}/${timestamp()}",
    "roleArn": "arn:aws:iam::1111-1111-1111:role/my-iot-role"
   }
```

```
  }
 ]
}
```

Para obter mais informações sobre como definir uma ação do Amazon S3 em umaAWS IoTregra, consulteAWS IoTAções de regra do — Amazon S3.

Faça as tarefas da Conta B

1. ConfigureAWS CLIusando o usuário IAM da conta B.
2. Crie uma política de bucket que confia no principal da conta A.

   Veja a seguir um exemplo de arquivo de carga útil que define uma política de bucket que confia no principal de outra conta.

```
{
 "Version": "2012-10-17",
 "Statement": [
  {
   "Sid": "AddCannedAcl",
   "Effect": "Allow",
   "Principal": {
    "AWS": [
     "arn:aws:iam::1111-1111-1111:root"
    ]
   },
   "Action": "s3:PutObject",
   "Resource": "arn:aws:s3:::ExampleBucket/*"
  }
 ]
}
```

   Para obter mais informações, consulteExemplos de políticas de bucket.
3. Para anexar a política de bucket ao bucket especificado, execute ocomando put-bucket-policy.

```
aws s3api put-bucket-policy --bucket ExampleBucket --policy file://./my-bucket-policy.json
```

4. Para que o acesso entre contas funcione, verifique se você tem o corretoBlock all public access (Bloquear todo o acesso público)Configurações do . Para obter mais informações, consulteMelhores práticas de segurança para o Amazon S3.

Configuração entre contas paraAWS Lambda

Cenário: A conta A invoca umAWS Lambdafunção da conta B, passando uma mensagem MQTT.

Conta da AWS	Conta referida como	Descrição
1111-1111-1111	Conta A	Ação da regra:`lambda:InvokeFunction`
2222-2222-2222	Conta B	ARN da função do Lambda. *arn:aws:lambda:region:2222-2222-2222:function:exampl function*

Faça as tarefas da Conta A

Observações

Para executar os seguintes comandos, o usuário do IAM deve ter permissões para`iot:CreateTopicRule`com ARN de regra como recurso e permissões para`iam:PassRole`ação com recurso como função ARN.

1. ConfigureAWS CLIusando o usuário IAM da conta A.
2. Execute acomando create-topic-rulepara criar uma regra que define o acesso entre contas à função Lambda da conta B.

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

Veja a seguir um exemplo de arquivo de carga útil com uma regra que insere todas as mensagens enviadas para `aiot/test`tópico na função Lambda especificada. A declaração do SQL filtra as mensagens, e o ARN da função concedeAWS IoTA permissão para passar os dados para a função do Lambda.

```
{
 "sql": "SELECT * FROM 'iot/test'",
 "ruleDisabled": false,
 "awsIotSqlVersion": "2016-03-23",
 "actions": [
  {
   "lambda": {
    "functionArn": "arn:aws:lambda:region:2222-2222-2222:function:example-function"
   }
  }
 ]
}
```

Para obter mais informações sobre como definir umAWS Lambdaação em umAWS IoTregra, leiaAWS IoTAções de regra do Lambda.

Faça as tarefas da Conta B

1. ConfigureAWS CLIusando o usuário IAM da conta B.
2. Execução docomando add-permission do LambdaPara darAWS IoTRegras de permissão para acionar a função do Lambda. Para executar o seguinte comando, o usuário do IAM deve ter permissão para`lambda:AddPermission`Ação .

```
aws lambda add-permission --function-name example-function --region us-east-1 --
principal iot.amazonaws.com --source-arn arn:aws:iot:region:1111-1111-1111:rule/
example-rule --source-account 1111-1111-1111 --statement-id "unique_id" --action
 "lambda:InvokeFunction"
```

Opções:

--principal

Este campo dá permissão paraAWS IoT(representado por`iot.amazonaws.com`) para chamar a função do Lambda.

--arn de origem

Esse campo garante que somente`arn:aws:iot:region:1111-1111-1111:rule/example-rule`emAWS IoTaciona esta função do Lambda e nenhuma outra regra na mesma conta ou diferente pode acionar essa função do Lambda.

--conta de origem

Esse campo garante queAWS IoTaciona esta função do Lambda somente em nome do`1111-1111-1111`conta.

Observações

Se você vir uma mensagem de erro “A regra não pôde ser encontrada” do seuAWS Lambdaconsole da função sobConfiguração, ignore a mensagem de erro e prossiga para testar a conexão.

Tratamento de erros (ação de erro)

Quando a AWS IoT recebe uma mensagem de um dispositivo, o mecanismo de regras verifica se a mensagem corresponde a uma regra. Se corresponder, a instrução de consulta da regra será avaliada e as ações da regra serão acionadas, passando o resultado da instrução de consulta.

Se ocorrer um problema ao acionar uma ação, o mecanismo de regras acionará uma ação de erro, se uma ação estiver especificada para a regra. Isso pode acontecer quando:

- Uma regra não tem permissão para acessar um bucket do Amazon S3.
- Um erro do usuário faz com que o throughput provisionado pelo DynamoDB seja excedido.

Formato da mensagem de ação de erro

Uma única mensagem é gerada por regra e mensagem. Por exemplo, se houver uma falha em duas ações de regra na mesma regra, a ação de erro receberá uma mensagem contendo os dois erros.

A mensagem de ação de erro parece com o exemplo a seguir.

```
{
  "ruleName": "TestAction",
  "topic": "testme/action",
  "cloudwatchTraceId": "7e146a2c-95b5-6caf-98b9-50e3969734c7",
  "clientId": "iotconsole-1511213971966-0",
  "base64OriginalPayload": "ewogICJtZXNzYWdlIjogIkhlbGxvIHZyb20gQVdTIElvVCBjb25zb2xlIgp9",
  "failures": [
    {
      "failedAction": "S3Action",
      "failedResource": "us-east-1-s3-verify-user",
      "errorMessage": "Failed to put S3 object. The error received was The
 specified bucket does not exist (Service: Amazon S3; Status Code: 404; Error
 Code: NoSuchBucket; Request ID: 9DF5416B9B47B9AF; S3 Extended Request ID:
 yMah1cwPhqTH267QLPhTKeVPKJB8BO5ndBHzOmWtxLTM6uAvwYYuqieAKyb6qRPTxP1tHXCoR4Y=).
 Message arrived on: error/action, Action: s3, Bucket: us-
east-1-s3-verify-user, Key: \"aaa\". Value of x-amz-id-2:
 yMah1cwPhqTH267QLPhTKeVPKJB8BO5ndBHzOmWtxLTM6uAvwYYuqieAKyb6qRPTxP1tHXCoR4Y="
    }
  ]
}
```

ruleName

O nome da regra que acionou a ação de erro.

tópico

O tópico no qual a mensagem original foi recebida.

cloudwatchTraceId

Uma identidade exclusiva que faz referência aos logs de erro no CloudWatch.

clientId

O ID do cliente do publicador da mensagem.

base64OriginalPayload

A carga da mensagem original codificada em Base64.

falhas

failedAction

O nome da ação que não foi concluída (por exemplo "S3Action").

failedResource

O nome do recurso (por exemplo, o nome de um bucket do S3).

errorMessage

A descrição e a explicação do erro.

Exemplo de ação de erro

Este é um exemplo de uma regra com uma ação de erro adicional. A seguinte regra tem uma ação que grava dados de mensagens em uma tabela do DynamoDB e uma ação de erro que grava dados em um bucket do Amazon S3:

```
{
    "sql" : "SELECT * FROM ..."
    "actions" : [{
        "dynamoDB" : {
            "table" : "PoorlyConfiguredTable",
            "hashKeyField" : "AConstantString",
            "hashKeyValue" : "AHashKey"}}
    ],
    "errorAction" : {
        "s3" : {
            "roleArn": "arn:aws:iam::123456789012:role/aws_iot_s3",
            "bucketName" : "message-processing-errors",
            "key" : "${replace(topic(), '/', '-') + '-' + timestamp() + '-' + newuuid()}"
        }
    }
}
```

Você pode usar qualquer função ou substituição em uma instrução SQL de ação de erro, exceto para funções externas (por exemplo, `get_thing_shadow`, `aws_lambda` e `machinelearning_predict`).

Para obter mais informações sobre regras e como especificar uma ação de erro, consulte Criar uma regra da AWS IoT.

Para obter mais informações sobre como usar CloudWatch Para monitorar o sucesso ou a falha de regras, consulteMétricas e dimensões do AWS IoT (p. 443).

Reduzir custos do sistema de mensagens com a Ingestão básica

Com o Basic Ingest, você pode enviar dados do dispositivo com segurança para oAWSserviços compatíveis comAções de regra do AWS IoT (p. 485), sem incorrerda mensagem. A Ingestão básica otimiza o fluxo de dados removendo o agente de mensagens de publicação/assinatura do caminho de ingestão, tornando-o mais econômico.

Você usa o Basic Ingest para enviar mensagens de seus dispositivos ou aplicativos. As mensagens têm nomes de tópicos que começam com`$aws/rules/`*`rule_name`*para seus três primeiros níveis, onde*`rule_name`*É o nome doAWS IoTregra que você deseja que o invoque.

Você pode continuar a usar uma regra existente com Ingestão básica adicionando o prefixo de Ingestão básica (`$aws/rules/`*`rule_name`*) para o tópico da mensagem que você normalmente usaria para invocar a regra. Por exemplo, se você tiver uma regra chamada`BuildingManager`que é invocado por mensagens com tópicos como`Buildings/Building5/Floor2/Room201/Lights`(`"sql": "SELECT * FROM 'Buildings/#'"`), você pode invocar a mesma regra com Ingestão básica enviando uma mensagem com o tópico`$aws/rules/BuildingManager/Buildings/Building5/Floor2/Room201/Lights`.

Observação:

- Seus dispositivos e regras não podem se inscrever em tópicos reservados de Ingestão básica. Para obter mais informações, consulte Tópicos reservados (p. 101).
- Se você precisar de um agente de publicação/assinatura para distribuir mensagens a vários assinantes (por exemplo, para entregar mensagens para outros dispositivos e para o mecanismo de regras), deverá continuar a usar o agente de mensagens do AWS IoT para lidar com a distribuição da mensagem. No entanto, certifique-se de publicar suas mensagens em tópicos que não sejam tópicos de Ingestão básica.

Usar a Ingestão básica

Antes de usar a Ingestão básica, verifique se o dispositivo ou aplicativo está usando umregra (p. 332)que tem permissões de publicação em`$aws/rules/*`. Como alternativa, você pode especificar permissões para regras individuais com`$aws/rules/`*`rule_name`*`/*`na política. Caso contrário, seus dispositivos e aplicativos poderão continuar a usar suas conexões existentes com o AWS IoT Core.

Quando a mensagem atinge o mecanismo de regras, não há diferença na execução ou no tratamento de erros entre as regras invocadas pela Ingestão básica e as chamadas pelas assinaturas do agente de mensagens.

Você pode criar regras para uso com Ingestão básica. Lembre-se do seguinte:

- O prefixo inicial de um tópico de Ingestão básica (`$aws/rules/`*`rule_name`*) não está disponível na função topic(Decimal) (p. 619).
- Se você definir uma regra invocada apenas com Ingestão básica, o`FROM`cláusula é opcional na`sql`campo do`rule`definição. Ela ainda será necessária se a regra também for invocada por outras mensagens que devem ser enviadas por meio do agente de mensagens (por exemplo, porque essas outras mensagens devem ser distribuídas a vários assinantes). Para obter mais informações, consulte Referência SQL do AWS IoT (p. 568).
- Os três primeiros níveis do tópico de Ingestão básica (`$aws/rules/`*`rule_name`*) não são contados em direção ao limite de comprimento de 8 segmentos ou no limite total de 256 caracteres de um tópico. Caso contrário, as mesmas restrições se aplicarão, conforme documentado emAWS IoTLimites.

- Se uma mensagem for recebida com um tópico de Ingestão básica que especifica uma regra inativa ou uma regra que não existe, um log de erros será criado em um log do Amazon CloudWatch para ajudá-lo com a depuração. Para obter mais informações, consulte Entradas de log do mecanismo de regras (p. 462). Uma métrica `RuleNotFound` é indicada e você pode criar alarmes nessa métrica. Para obter mais informações, consulte Métricas de regras em Métricas de regra (p. 444).
- Você ainda poderá publicar com QoS 1 em tópicos de Ingestão básica. Você recebe uma PUBACK depois que a mensagem é entregue com êxito ao mecanismo de regras. O recebimento de uma PUBACK não significa que as ações de regra foram concluídas com êxito. Você pode configurar uma ação de erro para lidar com erros quando uma ação for executada. Para obter mais informações, consulte Tratamento de erros (ação de erro) (p. 565).

Referência SQL do AWS IoT

Na AWS IoT, as regras são definidas usando uma sintaxe do tipo SQL. As declarações do SQL são compostas por três tipos de cláusulas:

SELECT

Obrigatório. Extrai informações da carga útil de uma mensagem recebida e executa transformações nas informações. As mensagens a serem usadas são identificadas peloFiltro de tópicos (p. 100)especificado na cláusula FROM.

A cláusula SELECT suportaTipos de dados (p. 572),Operadores (p. 575),Funções (p. 581),Literais (p. 624),Declarações de caso (p. 625),Extensões JSON (p. 626),Modelos de substituição (p. 627),Consultas de objeto aninhado (p. 629), eCargas binárias (p. 630).

FROM

A mensagem MQTTFiltro de tópicos (p. 100)que identifica as mensagens das quais extrair dados. A regra é acionada para cada mensagem enviada para um tópico MQTT que corresponda ao filtro de tópico especificado aqui. Obrigatório para regras que são acionadas por mensagens que passam pelo operador de mensagens. Opcional para regras que só são acionadas usando o recurso Ingestão básica (p. 567).

WHERE

(Opcional) Adiciona lógica condicional que determina se as ações especificadas por uma regra são executadas.

A cláusula WHERE é compatível com Tipos de dados (p. 572), Operadores (p. 575), Funções (p. 581), Literais (p. 624), Declarações de caso (p. 625), Extensões JSON (p. 626), Modelos de substituição (p. 627) e Consultas de objeto aninhado (p. 629).

Um exemplo de declaração do SQL é semelhante a:

```
SELECT color AS rgb FROM 'topic/subtopic' WHERE temperature > 50
```

Um exemplo de mensagem MQTT (também chamada de carga útil de entrada) é semelhante a:

```
{
    "color":"red",
    "temperature":100
}
```

Se essa mensagem for publicada no tópico `'topic/subtopic'`, a regra será acionada, e a declaração do SQL será avaliada. A declaração do SQL extrairá o valor da propriedade `color` se a propriedade

`"temperature"` for superior a 50. A cláusula WHERE especifica a condição `temperature > 50`. A palavra-chave `AS` renomeia a propriedade `"color"` para `"rgb"`. O resultado (também chamado de carga útil de saída) é semelhante a:

```
{
    "rgb":"red"
}
```

Esses dados são encaminhados para a ação da regra, que envia os dados para realizar mais processamento. Para obter mais informações sobre as ações de regra, consulte Ações de regra do AWS IoT (p. 485).

> **Note**
>
> No momento, os comentários não são compatíveis com oAWS IoTSintaxe SQL.
> Nomes de atributos com espaços neles não podem ser usados como nomes de campo na instrução SQL. Embora a carga recebida possa ter nomes de atributos com espaços neles, esses nomes não podem ser usados na instrução SQL. No entanto, eles serão passados para a carga útil de saída se você usar uma especificação de nome de campo curinga (*).

Cláusula SELECT

A cláusula SELECT do AWS IoT é essencialmente a mesma que a cláusula SELECT do SQL padrão ANSI, com algumas pequenas diferenças.

A cláusula SELECT suportaTipos de dados (p. 572),Operadores (p. 575),Funções (p. 581),Literais (p. 624),Declarações de caso (p. 625),Extensões JSON (p. 626),Modelos de substituição (p. 627),Consultas de objeto aninhado (p. 629), eCargas binárias (p. 630).

Você pode usar a cláusula SELECT para extrair informações das mensagens MQTT recebidas. Você também pode usar `SELECT *` para recuperar toda a carga útil da mensagem recebida. Por exemplo:

```
Incoming payload published on topic 'topic/subtopic': {"color":"red", "temperature":50}
SQL statement: SELECT * FROM 'topic/subtopic'
Outgoing payload: {"color":"red", "temperature":50}
```

Se a carga útil for um objeto JSON, você poderá fazer referência a chaves no objeto. A carga útil de saída contém o par de chave/valor. Por exemplo:

```
Incoming payload published on topic 'topic/subtopic': {"color":"red", "temperature":50}
SQL statement: SELECT color FROM 'topic/subtopic'
Outgoing payload: {"color":"red"}
```

Você pode usar o teclado AS para renomear chaves. Por exemplo:

```
Incoming payload published on topic 'topic/subtopic':{"color":"red", "temperature":50}
SQL:SELECT color AS my_color FROM 'topic/subtopic'
Outgoing payload: {"my_color":"red"}
```

Você pode selecionar vários itens ao separá-los com uma vírgula. Por exemplo:

```
Incoming payload published on topic 'topic/subtopic': {"color":"red", "temperature":50}
SQL: SELECT color as my_color, temperature as fahrenheit FROM 'topic/subtopic'
Outgoing payload: {"my_color":"red","fahrenheit":50}
```

Você pode selecionar vários itens incluindo '*' para adicionar itens à carga útil de entrada. Por exemplo:

```
Incoming payload published on topic 'topic/subtopic': {"color":"red", "temperature":50}
SQL: SELECT *, 15 as speed FROM 'topic/subtopic'
Outgoing payload: {"color":"red", "temperature":50, "speed":15}
```

Você pode usar a palavra-chave `"VALUE"` para produzir cargas úteis de saída que não são objetos JSON. Com a versão SQL `2015-10-08`, você pode selecionar apenas um item. Com a versão `2016-03-23` do SQL ou posterior, você também pode selecionar uma matriz para saída como um objeto de nível superior.

Example

```
Incoming payload published on topic 'topic/subtopic': {"color":"red", "temperature":50}
SQL: SELECT VALUE color FROM 'topic/subtopic'
Outgoing payload: "red"
```

Você pode usar a sintaxe `'.'` para analisar os objetos JSON aninhados na carga útil de entrada. Por exemplo:

```
Incoming payload published on topic 'topic/subtopic': {"color":
{"red":255,"green":0,"blue":0}, "temperature":50}
SQL: SELECT color.red as red_value FROM 'topic/subtopic'
Outgoing payload: {"red_value":255}
```

Para obter informações sobre como usar nomes de objetos e propriedades JSON que incluem caracteres reservados, como números ou o caractere hífen (menos), consulteExtensões JSON (p. 626)

Você pode usar funções (consulte Funções (p. 581)) para transformar a carga útil de entrada. Você pode usar parênteses para agrupamento. Por exemplo:

```
Incoming payload published on topic 'topic/subtopic': {"color":"red", "temperature":50}
SQL: SELECT (temperature - 32) * 5 / 9 AS celsius, upper(color) as my_color FROM 'topic/
subtopic'
Outgoing payload: {"celsius":10,"my_color":"RED"}
```

Cláusula FROM

A cláusula FROM inscreve sua regra em umtópico (p. 99)ouFiltro de tópicos (p. 100). Você deve colocar o filtro de tópicos ou tópico entre aspas simples ('). A regra é acionada para cada mensagem enviada para um tópico MQTT que corresponda ao filtro de tópico especificado aqui. Um filtro de tópicos permite se inscrever em um grupo de tópicos semelhantes.

Exemplo:

Carga útil de entrada publicada no tópico `'topic/subtopic'`: `{temperature: 50}`

Carga útil de entrada publicada no tópico `'topic/subtopic-2'`: `{temperature: 50}`

SQL: `"SELECT temperature AS t FROM 'topic/subtopic'"`.

A regra é inscrita em `'topic/subtopic'`, de modo que a carga útil de entrada é passada para a regra. A carga útil de saída, passada para as ações da regra, é: `{t: 50}`. A regra não está inscrita `'topic/subtopic-2'`, portanto, a regra não é acionada para a mensagem publicada em `'topic/subtopic-2'`.

Exemplo de curinga #:

Você pode usar o caractere curinga "#" (com vários níveis) para corresponder a um ou mais elementos de caminho específicos:

Carga útil de entrada publicada no tópico `'topic/subtopic'`: `{temperature: 50}`.

Carga útil de entrada publicada no tópico `'topic/subtopic-2'`: `{temperature: 60}`.

Carga útil de entrada publicada no tópico `'topic/subtopic-3/details'`: `{temperature: 70}`.

Carga útil de entrada publicada no tópico `'topic-2/subtopic-x'`: `{temperature: 80}`.

SQL: `"SELECT temperature AS t FROM 'topic/#'"`.

A regra é inscrita em qualquer tópico que comece com `'topic'`, por isso é executado três vezes, enviando cargas úteis de saída de `{t: 50}`(para tópico/subtópico), `{t: 60}`(para tópico/subtópico-2), e `{t: 70}`(para tópico/subtópico-3/detalhes) para suas ações. Se não estiver inscrito em `'topic-2/subtopic-x'`, a regra não será acionada para a mensagem `{temperature: 80}`.

Exemplo de curinga +:

Você pode usar o caractere curinga "+" (com nível único) para corresponder a qualquer elemento de caminho específico:

Carga útil de entrada publicada no tópico `'topic/subtopic'`: `{temperature: 50}`.

Carga útil de entrada publicada no tópico `'topic/subtopic-2'`: `{temperature: 60}`.

Carga útil de entrada publicada no tópico `'topic/subtopic-3/details'`: `{temperature: 70}`.

Carga útil de entrada publicada no tópico `'topic-2/subtopic-x'`: `{temperature: 80}`.

SQL: `"SELECT temperature AS t FROM 'topic/+'"`.

A regra é inscrita em todos os tópicos com dois elementos de caminho, em que o primeiro elemento é `'topic'`. A regra é executada para as mensagens enviadas para `'topic/subtopic'` e `'topic/subtopic-2'`, mas não `'topic/subtopic-3/details'`(tem mais níveis do que o filtro de tópicos) ou `'topic-2/subtopic-x'`(não começa com `topic`).

Cláusula WHERE

A cláusula WHERE determina se as ações especificadas por uma regra são executadas. Se a cláusula WHERE for avaliada como verdadeira, as ações da regra serão executadas. Caso contrário, as ações da regra não serão executadas.

A cláusula WHERE é compatível com Tipos de dados (p. 572), Operadores (p. 575), Funções (p. 581), Literais (p. 624), Declarações de caso (p. 625), Extensões JSON (p. 626), Modelos de substituição (p. 627) e Consultas de objeto aninhado (p. 629).

Exemplo:

Carga útil de entrada publicada em `topic/subtopic`: `{"color":"red", "temperature":40}`.

SQL: `SELECT color AS my_color FROM 'topic/subtopic' WHERE temperature > 50 AND color <> 'red'`.

Nesse caso, a regra será acionada, mas as ações especificadas pela regra não serão executadas. Não haverá carga útil de saída.

Você pode usar funções e operadores na cláusula WHERE. No entanto, você não pode fazer referência a aliases criados com a palavra-chave AS no SELECT. A cláusula WHERE é avaliada primeiro para determinar se SELECT será avaliada.

Exemplo com carga útil não JSON:

Carga útil não-JSON recebida publicada em `tópico/subtópico`: `80`

SQL:`SELECT decode(encode(*, 'base64'), 'base64') AS value FROM 'topic/subtopic' WHERE decode(encode(*, 'base64'), 'base64') > 50`

Nesse caso, a regra será acionada e as ações especificadas pela regra serão executadas. A carga útil de saída será transformada pela cláusula SELECT como uma carga útil JSON`{"value":80}`.

Tipos de dados

O mecanismo de regras da AWS IoT é compatível com todos os tipos de dados JSON.

Tipos de dados compatíveis

Type	Significado
`Int`	Um separado`Int`. Máximo de 34 dígitos.
`Decimal`	Um `Decimal` com uma precisão de 34 dígitos, com um mínimo de magnitude não zero de 1E-999 e um máximo de magnitude de 9,999...E999. Note Algumas funções retornam`Decimal`Valores com precisão dupla em vez de precisão de 34 dígitos. Com SQL V2 (2016-03-23), valores numéricos que são números inteiros, como`10.0`, são processados como um`Int`valor (`10`) em vez do esperado`Decimal`valor (`10.0`). Para processar valores numéricos de número inteiro de forma confiável como`Decimal`, use SQL V1 (2015-10-08) para a instrução de consulta de regra.
`Boolean`	`True` ou `False`.
`String`	Uma string UTF-8.
`Array`	Uma série de valores que não precisam ter o mesmo tipo.
`Object`	Um valor JSON que consiste em uma chave e um valor. As chaves devem ser strings. Os valores podem ser de qualquer tipo.
`Null`	`Null` conforme definido pelo JSON. É um valor real que representa a ausência de um valor. Você pode criar explicitamente um valor `Null` usando a palavra-chave `Null` na declaração do SQL. Por exemplo: `"SELECT NULL AS n FROM 'topic/subtopic'"`
`Undefined`	Não é um valor. Não é explicitamente representável no JSON, exceto ao omitir o valor. Por exemplo, no objeto `{"foo": null}`,

Type	Significado

	a chave "foo" gera NULL, mas a chave "bar" gera `Undefined`. Internamente, a linguagem SQL trata `Undefined` como um valor, mas não é representável no JSON; portanto, quando serializados para JSON, os resultados são `Undefined`.
	`{"foo":null, "bar":undefined}`
	é serializado para JSON como:
	`{"foo":null}`
	Da mesma forma, `Undefined` é convertido em uma string vazia quando serializado por conta própria. Funções chamadas com argumentos inválidos (por exemplo, tipos incorretos, número incorreto de argumentos, etc.) retornam `Undefined`.

Conversões

A tabela a seguir indica os resultados quando um valor de um tipo é convertido em outro tipo (quando um valor do tipo incorreto é dado a uma função). Por exemplo, se a função de valor absoluto "abs" (que espera um `Int` ou `Decimal`) for dado a `String`, ela tentará converter `String` em um `Decimal`, seguindo essas regras. Nesse caso, 'abs ("-5,123")' é tratado como "abs(-5,123)'.

> **Note**
>
> Não há tentativa de conversões para `Array`, `Object`, `Null` ou `Undefined`.

Para decimal

Tipo de argumento	Result
`Int`	Um `Decimal` sem ponto decimal.
`Decimal`	O valor de origem.
`Boolean`	`Undefined`. (Você pode explicitamente usar a função cast para transformar verdadeiro = 1,0, falso = 0,0.)
`String`	O mecanismo do SQL tenta analisar a string como um `Decimal`. A AWS IoT tenta analisar as strings correspondendo à expressão regular: `^-?\d+(\.\d+)?((?i)E-?\d+)?$`. "0", "-1,2" e "5E-12" são exemplos de strings que são automaticamente convertidas em `Decimals`.
Array	`Undefined`.
Objeto	`Undefined`.
Nulo	`Null`.

Tipo de argumento	Result
Não definido	`Undefined`.

Para int

Tipo de argumento	Result
`Int`	O valor de origem.
`Decimal`	O valor de origem arredondado para o `Int` mais próximo.
`Boolean`	`Undefined`. (Você pode explicitamente usar a função cast para transformar verdadeiro = 1,0, falso = 0,0.)
`String`	O mecanismo do SQL tenta analisar a string como um `Decimal`. A AWS IoT tenta analisar as strings correspondendo à expressão regular: `^-?\d+(\.\d+)?((?i)E-?\d+)?$`. "0", "-1,2"e "5E-" são exemplos de strings que são automaticamente convertidas em `Decimals`. A AWS IoT tenta converter `String` em um `Decimal` e, depois, trunca as casas decimais desse `Decimal` para fazer um `Int`.
Array	`Undefined`.
Objeto	`Undefined`.
Nulo	`Null`.
Não definido	`Undefined`.

Para Booleano

Tipo de argumento	Result
`Int`	`Undefined`. (Você pode usar explicitamente `ocast`função para transformar 0 = Falso, qualquer_valor_não_zero = Verdadeiro.)
`Decimal`	`Undefined`. (Você pode explicitamente usar a função cast para transformar 0 = Falso, qualquer_valor_não_zero = Verdadeiro.)
`Boolean`	O valor original.
`String`	"true" = Verdadeiro e "false" = Falso (não diferencia maiúsculas de minúsculas). Outros valores de string são `Undefined`.
Array	`Undefined`.
Objeto	`Undefined`.
Nulo	`Undefined`.

Tipo de argumento	Result
Não definido	`Undefined`.

Para string

Tipo de argumento	Result
`Int`	Uma representação de string do `Int` em notação padrão.
`Decimal`	Uma string representando o valor `Decimal`, possivelmente em notação científica.
`Boolean`	"true" ou "false". Todas as letras minúsculas.
`String`	O valor original.
Array	A `Array` serializada para JSON. A string resultante é uma lista separada por vírgulas, entre colchetes. A `String` tem aspas. Um `Decimal`, `Int`, `Boolean` e `Null` não.
Objeto	O objeto serializado para JSON. A string resultante é uma lista separada por vírgulas de pares de chave/valor e começa e termina com chaves. A `String` tem aspas. Um `Decimal`, `Int`, `Boolean` e `Null` não.
Nulo	`Undefined`.
Não definido	Indefinido.

Operadores

Os operadores a seguir podem ser usados nas cláusulas SELECT e WHERE.

Operador AND

Gera um resultado `Boolean`. Realiza uma operação E lógica. Retornará verdadeiro se os operandos esquerdo e direito forem verdadeiros. Caso contrário, retorna false. Os operandos `Boolean` ou operandos de string "true" ou "false" que não diferenciam letras maiúsculas de minúsculas são necessários.

Sintaxe: *`expression`* `AND` *`expression`*.

Operador AND

Operando esquerdo	Operando direito	Resultado
`Boolean`	`Boolean`	`Boolean`. Verdadeiro se ambos os operandos forem verdadeiros. Caso contrário, falso.
`String/Boolean`	`String/Boolean`	Se todas as strings forem "true" ou "false" (não diferencia letras maiúsculas de minúsculas), elas serão convertidas em `Boolean` e processadas normalmente como *`boolean`* `AND` *`boolean`*.

Operando esquerdo	Operando direito	Resultado
Outros valores	Outros valores	`Undefined`.

Operador OR

Gera um resultado `Boolean`. Executa uma operação OU lógica. Retornará verdadeiro se um dos operandos esquerdo ou direito for verdadeiro. Caso contrário, retorna false. Os operandos `Boolean` ou operandos de string "true" ou "false" que não diferenciam letras maiúsculas de minúsculas são necessários.

Sintaxe: *`expression`* `OR` *`expression`*.

Operador OR

Operando esquerdo	Operando direito	Resultado
`Boolean`	`Boolean`	`Boolean`. Verdadeiro se um dos operandos for verdadeiro. Caso contrário, falso.
`String`/`Boolean`	`String`/`Boolean`	Se todas as strings forem "true" ou "false" (não diferencia letras maiúsculas de minúsculas), elas serão convertidas em booleanos e processadas normalmente como *`boolean`* `OR` *`boolean`*.
Outros valores	Outros valores	`Undefined`.

Operador NOT

Gera um resultado `Boolean`. Realiza uma operação NÃO lógica. Retornará verdadeiro se o operando for falso. Caso contrário, retornará verdadeiro. Um operando `Boolean` ou operando de string "true" ou "false" que não diferencia letras maiúsculas de minúsculas é necessário.

Sintaxe: `NOT` *`expression`*.

Operador NOT

Operando	Resultado
`Boolean`	`Boolean`. Verdadeiro se o operando for falso. Caso contrário, verdadeiro.
`String`	Se a string for "true" ou "false" (não diferencia letras maiúsculas de minúsculas), ela será convertida no valor booliano correspondente, e o valor oposto será gerado.
Outros valores	`Undefined`.

> operador

Gera um resultado `Boolean`. Verdadeiro se o operando esquerdo for superior ao operando direito. Os dois operandos são convertidos em um `Decimal` e depois comparados.

Sintaxe: `expression > expression`.

> operador

Operando esquerdo	Operando direito	Resultado
`Int/Decimal`	`Int/Decimal`	`Boolean`. Verdadeiro se o operando esquerdo for superior ao operando direito. Caso contrário, falso.
`String/Int/Decimal`	`String/Int/Decimal`	Se todas as strings puderem ser convertidas em `Decimal`, então `Boolean`. Verdadeiro se o operando esquerdo for superior ao operando direito. Caso contrário, falso.
Outros valores	`Undefined`.	`Undefined`.

>= operador

Gera um resultado `Boolean`. Verdadeiro se o operando esquerdo for superior ou igual ao operando direito. Os dois operandos são convertidos em um `Decimal` e depois comparados.

Sintaxe: `expression >= expression`.

>= operador

Operando esquerdo	Operando direito	Resultado
`Int/Decimal`	`Int/Decimal`	`Boolean`. Verdadeiro se o operando esquerdo for igual ou superior ao operando direito. Caso contrário, falso.
`String/Int/Decimal`	`String/Int/Decimal`	Se todas as strings puderem ser convertidas em `Decimal`, então `Boolean`. Verdadeiro se o operando esquerdo for superior ou igual ao operando direito. Caso contrário, falso.
Outros valores	`Undefined`.	`Undefined`.

< operador

Gera um resultado `Boolean`. Verdadeiro se o operando esquerdo for inferior ao operando direito. Os dois operandos são convertidos em um `Decimal` e depois comparados.

Sintaxe: `expression < expression`.

< operador

Operando esquerdo	Operando direito	Resultado
`Int/Decimal`	`Int/Decimal`	`Boolean`. Verdadeiro se o operando esquerdo for inferior ao operando direito. Caso contrário, falso.
`String/Int/Decimal`	`String/Int/Decimal`	Se todas as strings puderem ser convertidas em `Decimal`, então `Boolean`. Verdadeiro se o operando esquerdo for inferior ao operando direito. Caso contrário, falso.

Operando esquerdo	Operando direito	Resultado
Outros valores	`Undefined`	`Undefined`

<= operador

Gera um resultado `Boolean`. Verdadeiro se o operando esquerdo for inferior ou igual ao operando direito. Os dois operandos são convertidos em um `Decimal` e depois comparados.

Syntaxe: *`expression`* `<=` *`expression`*.

<= operador

Operando esquerdo	Operando direito	Resultado
`Int/Decimal`	`Int/Decimal`	`Boolean`. Verdadeiro se o operando esquerdo for igual ou inferior ao operando direito. Caso contrário, falso.
`String/Int/Decimal`	`String/Int/Decimal`	Se todas as strings puderem ser convertidas em `Decimal`, então `Boolean`. Verdadeiro se o operando esquerdo for inferior ou igual ao operando direito. Caso contrário, falso.
Outros valores	`Undefined`	`Undefined`

<> operador

Gera um resultado `Boolean`. Retornará verdadeiro se os operandos esquerdo e direito forem diferentes. Caso contrário, retornará false.

Sintaxe: *`expression`* `<>` *`expression`*.

<> operador

Operando esquerdo	Operando direito	Resultado
`Int`	`Int`	Verdadeiro se o operando esquerdo não for igual ao operando direito. Caso contrário, falso.
`Decimal`	`Decimal`	Verdadeiro se o operando esquerdo não for igual ao operando direito. Caso contrário, falso. `Int` é convertido em `Decimal` antes de ser comparado.
`String`	`String`	Verdadeiro se o operando esquerdo não for igual ao operando direito. Caso contrário, falso.
Array	Array	Verdadeiro se os itens em cada operando não forem iguais e não estiverem na mesma ordem. Caso contrário, falso
Objeto	Objeto	Verdadeiro se as chaves e os valores de cada operando não forem iguais. Caso contrário, falso. A ordem das chaves/valores é irrelevante.
Nulo	Nulo	Falso.
Qualquer valor	`Undefined`	Indefinido.

Operando esquerdo	Operando direito	Resultado
`Undefined`	Qualquer valor	Indefinido.
Tipo não correspondente	Tipo não correspondente	Verdadeiro.

= operador

Gera um resultado `Boolean`. Retornará verdadeiro se os operandos esquerdo e direito forem iguais. Caso contrário, retornará false.

Sintaxe: *`expression`* `=` *`expression`*.

= operador

Operando esquerdo	Operando direito	Resultado
`Int`	`Int`	Verdadeiro se o operando esquerdo for igual ao operando direito. Caso contrário, falso.
`Decimal`	`Decimal`	Verdadeiro se o operando esquerdo for igual ao operando direito. Caso contrário, falso. `Int` é convertido em `Decimal` antes de ser comparado.
`String`	`String`	Verdadeiro se o operando esquerdo for igual ao operando direito. Caso contrário, falso.
Array	Array	Verdadeiro se os itens em cada operando forem iguais e estiverem na mesma ordem. Caso contrário, falso.
Objeto	Objeto	Verdadeiro se as chaves e os valores de cada operando forem iguais. Caso contrário, falso. A ordem das chaves/valores é irrelevante.
Qualquer valor	`Undefined`	`Undefined`.
`Undefined`	Qualquer valor	`Undefined`.
Tipo não correspondente	Tipo não correspondente	Falso.

+ operador

" +" é um operador sobrecarregado. Ele pode ser usado para concatenação ou adição de string.

Sintaxe: *`expression`* `+` *`expression`*.

+ operador

Operando esquerdo	Operando direito	Resultado
`String`	Qualquer valor	Converte o operando direito em uma string e concatena-o ao final do operando esquerdo.

Operando esquerdo	Operando direito	Resultado
Qualquer valor	`String`	Converte o operando esquerdo em uma string e concatena o operando direito ao final do operando esquerdo convertido.
`Int`	`Int`	`Int` value. Adiciona operandos juntos.
`Int/Decimal`	`Int/Decimal`	`Decimal` value. Adiciona operandos juntos.
Outros valores	Outros valores	`Undefined`.

- operador

Subtrai o operando direito do operando esquerdo.

Sintaxe: *`expression`* `-` *`expression`*.

- operador

Operando esquerdo	Operando direito	Resultado
`Int`	`Int`	`Int` value. Subtrai o operando direito do operando esquerdo.
`Int/Decimal`	`Int/Decimal`	`Decimal` value. Subtrai o operando direito do operando esquerdo.
`String/Int/Decimal`	`String/Int/Decimal`	Se todas as strings forem convertidas em decimais corretamente, um valor `Decimal` será gerado. Subtrai o operando direito do operando esquerdo. Caso contrário, gera `Undefined`.
Outros valores	Outros valores	`Undefined`.
Outros valores	Outros valores	`Undefined`.

* operador

Multiplica o operando esquerdo pelo operando direito.

Sintaxe: *`expression`* `*` *`expression`*.

* operador

Operando esquerdo	Operando direito	Resultado
`Int`	`Int`	`Int` value. Multiplica o operando esquerdo pelo operando direito.
`Int/Decimal`	`Int/Decimal`	`Decimal` value. Multiplica o operando esquerdo pelo operando direito.
`String/Int/Decimal`	`String/Int/Decimal`	Se todas as strings forem convertidas em decimais corretamente, um valor `Decimal` será gerado. Multiplica o operando esquerdo pelo operando direito. Caso contrário, gera `Undefined`.

Operando esquerdo	Operando direito	Resultado
Outros valores	Outros valores	`Undefined`.

/ operador

Divide o operando esquerdo pelo operando direito.

Sintaxe: *`expression`* `/` *`expression`*.

/ operador

Operando esquerdo	Operando direito	Resultado
`Int`	`Int`	`Int` value. Divide o operando esquerdo pelo operando direito.
`Int`/`Decimal`	`Int`/`Decimal`	`Decimal` value. Divide o operando esquerdo pelo operando direito.
`String`/`Int`/`Decimal`	`String`/`Int`/`Decimal`	Se todas as strings forem convertidas em decimais corretamente, um valor `Decimal` será gerado. Divide o operando esquerdo pelo operando direito. Caso contrário, gera `Undefined`.
Outros valores	Outros valores	`Undefined`.

% operador

Gera o restante da divisão do operando esquerdo pelo operando direito.

Sintaxe: *`expression`* `%` *`expression`*.

% operador

Operando esquerdo	Operando direito	Resultado
`Int`	`Int`	`Int` value. Gera o restante da divisão do operando esquerdo pelo operando direito.
`String`/`Int`/`Decimal`	`String`/`Int`/`Decimal`	Se todas as strings forem convertidas em decimais corretamente, um valor `Decimal` será gerado. Gera o restante da divisão do operando esquerdo pelo operando direito. Caso contrário, `Undefined`.
Outros valores	Outros valores	`Undefined`.

Funções

Você pode usar as seguintes funções integradas nas cláusulas SELECT ou WHERE das expressões do SQL.

abs(Decimal)

Gera o valor absoluto de um número. Compatível com a versão de 08/10/2015 do SQL e posteriores.

Exemplo: `abs(-5)` gera 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 convertida, o resultado será `Undefined`.
Array	`Undefined`.
Objeto	`Undefined`.
Nulo	`Undefined`.
Não definido	`Undefined`.

accountid()

Gera o ID da conta que possui essa regra como uma `String`. Compatível com a versão de 08/10/2015 do SQL e posteriores.

Exemplo:

`accountid()` = "123456789012"

acos(Decimal)

Gera o cosseno inverso de um número em radianos. Argumentos `Decimal` são arredondados para dobrar a precisão antes da aplicação da função. Compatível com a versão de 08/10/2015 do SQL e posteriores.

Exemplo: `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`, o cosseno inverso do argumento. Se a string não puder ser convertida, o resultado será `Undefined`. Os resultados imaginários são gerados como `Undefined`.

Tipo de argumento	Result
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. Compatível com a versão de 08/10/2015 do SQL e posteriores.

Exemplo: `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. Compatível com a versão de 08/10/2015 do SQL e posteriores.

Exemplo: `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`.

Tipo de argumento	Result
`Decimal`	`Decimal` (com precisão dupla), a tangente inversa do argumento. Os resultados imaginários são gerados como `Undefined`.
`Boolean`	`Undefined`.
`String`	`Decimal`, 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 (metade inferior, y < 0). Argumentos `Decimal` são arredondados para dobrar a precisão antes da aplicação da função. Compatível com a versão de 08/10/2015 do SQL e posteriores.

Exemplo: `atan2(1, 0)` = 1,5707963267948966

Tipo de argumento	Tipo de argumento	Result
`Int/Decimal`	`Int/Decimal`	`Decimal` (com precis ponto (x, y) especifica
`Int/Decimal/String`	`Int/Decimal/String`	`Decimal`, a tangente uma string não puder `Undefined`.
Outros valores	Outros valores	`Undefined`.

aws_lambda(functionArn, inputJson)

Chama a passagem da função Lambda especificada`inputJson`Para a função do Lambda e retorna o JSON gerado pela função do Lambda.

Argumentos

Argumento	Descrição
`functionArn`	O ARN da função Lambda a ser chamada. A função do Lambda deve retornar dados JSON.
`inputJson`	A entrada JSON passada para a função do Lambda. Para passar consultas de objetos aninhados e literais, é necessário usar a versão do SQL 2016-03-23.

Você deve concederAWS IoT `lambda:InvokeFunction`As permissões para invocar a função Lambda especificada. O seguinte exemplo mostra como conceder a permissão `lambda:InvokeFunction` usando a AWS CLI:

```
aws lambda add-permission --function-name "function_name"
--region "region"
--principal iot.amazonaws.com
--source-arn arn:aws:iot:us-east-1:account_id:rule/rule_name
--source-account "account_id"
--statement-id "unique_id"
--action "lambda:InvokeFunction"
```

Veja os argumentos do comando add-permission:

--nome da função

Nome da função Lambda. Você adiciona uma nova permissão para atualizar a política de recursos da função.

--região

ORegião da AWSda sua conta.

--principal

O principal que está recebendo a permissão. Isso deve ser`iot.amazonaws.com`permitirAWS IoTpermissão para chamar uma função do Lambda.

--arn de origem

O ARN da regra. Você pode usar oget-topic-rule AWS CLIComando para obter o ARN de uma regra.

--conta de origem

OConta da AWSonde a regra está definida.

--id da declaração

Um identificador de declaração exclusivo.

--action

A ação do Lambda que você deseja permitir nesta declaração. Para permitirAWS IoTPara chamar uma função do Lambda, especifique`lambda:InvokeFunction`.

> **Important**
>
> Se você adicionar uma permissão para umAWS IoTprincipal sem fornecer `source-arn`ou`source-account`, QualquerConta da AWSO que criar uma regra com sua ação do Lambda do pode acionar regras para invocar sua função do Lambda da daAWS IoT. Para obter mais informações, consulteModelo de permissão do Lambda.

Dada uma carga da mensagem JSON como:

```
{
    "attribute1": 21,
    "attribute2": "value"
}
```

O`aws_lambda`A função pode ser usada para chamar a função do Lambda da da seguinte forma.

```
SELECT
aws_lambda("arn:aws:lambda:us-east-1:account_id:function:lambda_function",
 {"payload":attribute1}) as output FROM 'topic-filter'
```

Se você deseja passar a carga da mensagem MQTT completa, especifique a carga JSON usando '*', como o exemplo a seguir.

```
SELECT
aws_lambda("arn:aws:lambda:us-east-1:account_id:function:lambda_function", *) as output
 FROM 'topic-filter'
```

`payload.inner.element`seleciona dados da mensagem publicada no tópico 'tópico/subtópico'.

`some.value`seleciona dados da saída gerada pela função do Lambda.

> **Note**
>
> O mecanismo de regras limita a duração da execução das funções do Lambda. As chamadas de função do Lambda em regras devem ser concluídas em 2.000 milissegundos.

bitand(Int, Int)

Realiza um bitwise AND nas representações do bit dos dois argumentos `Int`(-convertido). Compatível com a versão de 08/10/2015 do SQL e posteriores.

Exemplo: `bitand(13, 5)` = 5

Tipo de argumento	Tipo de argumento	Result
`Int`	`Int`	`Int`, um bitwise AND
`Int/Decimal`	`Int/Decimal`	`Int`, um bitwise AND números não `Int` são próximo. Se qualquer convertido em um `In`
`Int/Decimal/String`	`Int/Decimal/String`	`Int`, um bitwise AND strings são convertida para o `Int` mais próx resultado será `Undef`
Outros valores	Outros valores	`Undefined`.

bitor(Int, Int)

Executa um bitwise OR das representações de bit dos dois argumentos. Compatível com a versão de 08/10/2015 do SQL e posteriores.

Exemplo: `bitor(8, 5)` = 13

Tipo de argumento	Tipo de argumento	Result
`Int`	`Int`	`Int`, o bitwise OR dos
`Int/Decimal`	`Int/Decimal`	`Int`, o bitwise OR dos números não `Int` são próximo. Se não for p `Undefined`.
`Int/Decimal/String`	`Int/Decimal/String`	`Int`, o bitwise OR nos são convertidas em d

Tipo de argumento	Tipo de argumento	Result
		o `Int` mais próximo. S resultado será `Undef`
Outros valores	Outros valores	`Undefined`.

bitxor(Int, Int)

Realiza um bitwise XOR nas representações do bit dos dois argumentos `Int`(-convertido). Compatível com a versão de 08/10/2015 do SQL e posteriores.

Exemplo:`bitor(13, 5)` = 8

Tipo de argumento	Tipo de argumento	Result
`Int`	`Int`	`Int`, um bitwise XOR
`Int/Decimal`	`Int/Decimal`	`Int`, um bitwise XOR não `Int` são arredond
`Int/Decimal/String`	`Int/Decimal/String`	`Int`, um bitwise XOR convertidas em decim mais próximo. Se não será `Undefined`.
Outros valores	Outros valores	`Undefined`.

bitnot(Int)

Realiza um bitwise NOT nas representações do bit do argumento `Int`(-convertido). Compatível com a versão de 08/10/2015 do SQL e posteriores.

Exemplo: `bitnot(13)` = 2

Tipo de argumento	Result
`Int`	`Int`, um bitwise NOT do argumento.
`Decimal`	`Int`, um bitwise NOT do argumento. O valor `Decimal` é arredondado para o `Int` mais próximo.
`String`	`Int`, um bitwise NOT do argumento. As strings são convertidas em decimais e são arredondadas para o `Int` mais próximo. Se não for possível converter, o resultado será `Undefined`.
Outros valores	Outros valores.

cast()

Converte um valor de um tipo de dados para outro. O cast é muito parecido com as conversões padrão, com a adição da capacidade de converter números em ou de boolianos. Se a AWS IoT não puder determinar como converter de um tipo em outro, o resultado será `Undefined`. Compatível com a versão de 08/10/2015 do SQL e posteriores. Formato: cast(*`valor`* as *`tipo`*).

Exemplo:

`cast(true as Int) = 1`

As seguintes palavras-chave podem aparecer após "as" ao chamar `cast`:

Para as versões do SQL de 08/10/2015 e 23/03/2016

Palavra-chave	Result
`String`	Aplica valor para `String`.
Nvarchar	Aplica valor para `String`.
Texto	Aplica valor para `String`.
Ntext	Aplica valor para `String`.
varchar	Aplica valor para `String`.
`Int`	Aplica valor para `Int`.
Inteiro	Aplica valor para `Int`.
Double	Divergência de valor para`Decimal`(com precisão dupla).

Além disso, para as versões do SQL de 23/03/2016

Palavra-chave	Result
`Decimal`	Aplica valor para `Decimal`.
Bool	Aplica valor para `Boolean`.
`Boolean`	Aplica valor para `Boolean`.

Regras de cast:

Cast para decimal

Tipo de argumento	Result
`Int`	Um `Decimal` sem ponto decimal.
`Decimal`	O valor de origem. **Note** Com SQL V2 (2016-03-23), valores numéricos que são números inteiros, como`10.0`, retornar um`Int`valor (`10`) em vez do esperado`Decimal`valor (`10.0`). Para converter valores numéricos de número inteiro de forma confiável como`Decimal`, use SQL V1 (2015-10-08) para a instrução de consulta de regra.
`Boolean`	verdadeiro = 1,0, falso = 0,0.
`String`	Tenta analisar a string como um `Decimal`. A AWS IoT tenta analisar as strings correspondendo à regex: ^-?\d

Tipo de argumento	Result
	+(\.\d+)?((?i)E-?\d+)?$. "0", "-1,2" e "5E-12" são exemplos de strings que são automaticamente convertidas em decimais.
Array	`Undefined`.
Objeto	`Undefined`.
Nulo	`Undefined`.
Não definido	`Undefined`.

Cast para int

Tipo de argumento	Result
`Int`	O valor de origem.
`Decimal`	O valor de origem arredondado para o `Int` mais próximo.
`Boolean`	verdadeiro = 1,0, falso = 0,0.
`String`	Tenta analisar a string como um `Decimal`. A AWS IoT tenta analisar as strings correspondendo à regex: ^-?\d +(\.\d+)?((?i)E-?\d+)?$. "0", "-1,2", "5E-12" são exemplos de sstrings que são automaticamente convertidas em decimais. A AWS IoT tenta converter a string em `Decimal` e arredondar para o `Int` mais próximo.
Array	`Undefined`.
Objeto	`Undefined`.
Nulo	`Undefined`.
Não definido	`Undefined`.

Aplicar para **`Boolean`**

Tipo de argumento	Result
`Int`	0 = Falso, qualquer_valor_não_zero = Verdadeiro.
`Decimal`	0 = Falso, qualquer_valor_não_zero = Verdadeiro.
`Boolean`	O valor de origem.
`String`	"true" = Verdadeiro e "false" = Falso (não diferencia maiúsculas de minúsculas). Outros valores de string = `Undefined`.
Array	`Undefined`.
Objeto	`Undefined`.
Nulo	`Undefined`.
Não definido	`Undefined`.

Cast para string

Tipo de argumento	Result
`Int`	Uma representação de string do `Int` em notação padrão.
`Decimal`	Uma string representando o valor `Decimal`, possivelmente em notação científica.
`Boolean`	"verdadeiro" ou "falso", todas as letras minúsculas.
`String`	"true" = Verdadeiro e "false" = Falso (não diferencia maiúsculas de minúsculas). Outros valores de string = `Undefined`.
Array	A matriz serializada para JSON. A string resultante é uma lista separada por vírgulas, entre colchetes. `String` tem aspas. `Decimals`, `Ints` e `Booleans` não.
Objeto	O objeto serializado para JSON. A string JSON é uma lista separada por vírgulas de pares de chave/valor e começa e termina com chaves. `String` tem aspas. `Decimals`, `Ints`, `Booleans` e `Null` não.
Nulo	`Undefined`.
Não definido	`Undefined`.

ceil(Decimal)

Arredonda o `Decimal` fornecido para o `Int` mais próximo. Compatível com a versão de 08/10/2015 do SQL e posteriores.

Exemplos:

`ceil(1.2)` = 2

`ceil(-1.2)` = -1

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

chr(String)

Gera o caractere ASCII que corresponde ao dado argumento `Int`. Compatível com a versão de 08/10/2015 do SQL e posteriores.

Exemplos:

`chr(65)` = "A".

`chr(49)` = "1".

Tipo de argumento	Result
`Int`	O caractere correspondente ao valor ASCII especificado. Se o argumento não for um valor ASCII válido, o resultado será `Undefined`.
`Decimal`	O caractere correspondente ao valor ASCII especificado. O argumento `Decimal` é arredondado para o `Int` mais próximo. Se o argumento não for um valor ASCII válido, o resultado será `Undefined`.
`Boolean`	`Undefined`.
`String`	Se a `String` puder ser convertida em `Decimal`, ela será arredondada para o `Int` mais próximo. Se o argumento não for um valor ASCII válido, o resultado será `Undefined`.
Array	`Undefined`.
Objeto	`Undefined`.
Nulo	`Undefined`.
Outros valores	`Undefined`.

clientid()

Gera o ID do MQTT cliente que envia a mensagem ou `n/a` se a mensagem não foi enviada por MQTT. Compatível com a versão de 08/10/2015 do SQL e posteriores.

Exemplo:

`clientid()` = "123456789012"

concat()

Concatena as matrizes ou strings. Essa função aceita qualquer número de argumentos e gera uma `String` ou uma `Array`. Compatível com a versão de 08/10/2015 do SQL e posteriores.

Exemplos:

`concat()` = `Undefined`.

`concat(1)` = "1".

`concat([1, 2, 3], 4)` = [1, 2, 3, 4].

`concat([1, 2, 3], "hello")` = [1, 2, 3, "hello"]

`concat("con", "cat")` = "concat"

`concat(1, "hello")` = "1hello"

`concat("he","is","man")` = "heisman"

`concat([1, 2, 3], "hello", [4, 5, 6])` = [1, 2, 3, "hello", 4, 5, 6]

Número de argumentos	Resultado
0	`Undefined`.
1	O argumento é gerado sem modificações.
2+	Se algum argumento for uma `Array`, o resultado será uma única matriz que contém todos os argumentos. Se não houver matrizes e pelo menos um argumento for uma `String`, o resultado será a concatenação das representações de `String` de todos os argumentos. Os argumentos são convertidos em strings usando as conversões padrão indicadas acima. .

cos(Decimal)

Gera o cosseno de um número em radianos. Argumentos `Decimal` são arredondados para dobrar a precisão antes da aplicação da função. Compatível com a versão de 08/10/2015 do SQL e posteriores.

Exemplo:

`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. Compatível com a versão de 08/10/2015 do SQL e posteriores.

Exemplo: `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`.

decodificar (valor, decodingScheme)

Usar `adecode`função para decodificar um valor codificado. Se a string decodificada for um documento JSON, um objeto endereçável será retornado. Caso contrário, a string decodificada será retornada como uma string. A função retorna NULL se a string não puder ser decodificada.

Compatível com a versão de 23/03/2016 do SQL e posteriores.

value

Um valor de string ou qualquer uma das expressões válidas, conforme definido emReferência SQL do AWS IoT (p. 568), que retornam uma string.

Esquema de decoding

Uma string literal que representa o esquema usado para decodificar o valor. No momento, só há compatibilidade com `'base64'`.

Exemplo

Neste exemplo, a carga útil da mensagem inclui um valor codificado.

```
{
    encoded_temp: "eyAidGVtcGVyYXR1cmUiOiAzMyB9Cg=="
}
```

O`decode`nesta instrução SQL, decodifica o valor na carga útil da mensagem.

```
SELECT decode(encoded_temp,"base64").temperature AS temp from 'topic/subtopic'
```

Decodificação do`encoded_temp`o valor resulta no seguinte documento JSON válido, que permite que a instrução SELECT leia o valor da temperatura.

```
{ "temperature": 33 }
```

O resultado da declaração SELECT neste exemplo é mostrado aqui.

```
{ "temp": 33 }
```

Se o valor decodificado não fosse um documento JSON válido, o valor decodificado seria retornado como uma string.

encode(value, encodingScheme)

Use a função `encode` para codificar a carga útil, que provavelmente serão dados não JSON, em sua representação de string com base no esquema de codificação. Compatível com a versão de 23/03/2016 do SQL e posteriores.

value

Qualquer uma das expressões válidas, conforme definido em Referência SQL do AWS IoT (p. 568). Você pode especificar * para codificar toda a carga útil, independentemente de estar no formato JSON. Se você fornecer uma expressão, o resultado da avaliação será convertido em uma string antes de ser codificado.

encodingScheme

Uma string literal que representa o esquema de codificação que você deseja usar. No momento, só há compatibilidade com `'base64'`.

endswith(String, String)

Gera um `Boolean` indicando se o primeiro argumento de `String` termina com o segundo argumento de `String`. Se um dos argumentos for `Null` ou `Undefined`, o resultado será `Undefined`. Compatível com a versão de 08/10/2015 do SQL e posteriores.

Exemplo: `endswith("cat","at")` = verdadeiro.

Tipo de argumento 1	Tipo de argumento 2	Result
`String`	`String`	Verdadeiro se o prime argumento. Caso con
Outros valores	Outros valores	Os dois argumentos s as regras padrão de c argumento terminar n contrário, falso. Se un `Undefined`, o resulta

exp(Decimal)

Gera e elevado ao argumento `Decimal`. Argumentos `Decimal` são arredondados para dobrar a precisão antes da aplicação da função. Compatível com a versão de 08/10/2015 do SQL e posteriores.

Exemplo: `exp(1)` = e.

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

floor(Decimal)

Arredonda o `Decimal` fornecido para o `Int` mais próximo. Compatível com a versão de 08/10/2015 do SQL e posteriores.

Exemplos:

`floor(1.2)` = 1

`floor(-1.2)` = -2

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

get

Extrai um valor de um tipo de coleção (Matriz, String, Objeto). Nenhuma conversão é aplicada ao primeiro argumento. A conversão se aplica como documentado na tabela ao segundo argumento. Compatível com a versão de 08/10/2015 do SQL e posteriores.

Exemplos:

`get(["a", "b", "c"], 1)` = "b"

`get({"a":"b"}, "a")` = "b"

`get("abc", 1)`= “a”

Tipo de argumento 1	Tipo de argumento 2	Result
Array	Qualquer tipo (convertido em `Int`)	O item no índice com segundo argumento (possível converter, o índice estiver fora dos matriz.comprimento),

Tipo de argumento 1	Tipo de argumento 2	Result
String	Qualquer tipo (convertido em `Int`)	O caractere no índice pelo segundo argume for possível converter o índice estiver fora d string.comprimento),
Objeto	`String` (nenhuma conversão é aplicada)	O valor armazenado r correspondente à cha segundo argumento.
Outros valores	Qualquer valor	`Undefined`.

get_dynamodb(tableName, partitionKeyName, partitionKeyValue, sortKeyName, sortKeyValue, roleArn)

Recupera dados de uma tabela do DynamoDB.`get_dynamodb()`permite consultar uma tabela do DynamoDB enquanto uma regra é avaliada. É possível filtrar ou aumentar cargas de mensagens usando dados recuperados do DynamoDB. Compatível com a versão de 23/03/2016 do SQL e posteriores.

`get_dynamodb()` utiliza os seguintes parâmetros:

tableName

O nome da tabela do DynamoDB a ser consultada.

partitionKeyName

O nome da chave da partição. Para obter mais informações, consulteTeclas DynamoDB.

partitionKeyValue

O valor da chave de partição usada para identificar um registro. Para obter mais informações, consulteTeclas DynamoDB.

sortKeyName

(Opcional) O nome da chave de classificação. Esse parâmetro será necessário somente se a tabela do DynamoDB consultada usar uma chave composta. Para obter mais informações, consulteTeclas DynamoDB.

sortKeyValue

(Opcional) O valor da chave de classificação. Esse parâmetro será necessário somente se a tabela do DynamoDB consultada usar uma chave composta. Para obter mais informações, consulteTeclas DynamoDB.

roleArn

O ARN de uma função do IAM que concede acesso à tabela do DynamoDB. O mecanismo de regras assume essa função para acessar a tabela do DynamoDB em seu nome. Evite usar uma função excessivamente permissiva. Conceda à função apenas as permissões necessárias para a regra. Veja a seguir um exemplo de política que concede acesso a uma tabela do DynamoDB.

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

```
            "Action": "dynamodb:GetItem",
            "Resource": "arn:aws:dynamodb:aws-region:account-id:table/table-name"
        }
    ]
}}
```

Como um exemplo de como você pode usar`get_dynamodb()`, digamos que você tenha uma tabela do DynamoDB que contém o ID do dispositivo e as informações de localização para todos os dispositivos conectados aoAWS IoT. A instrução SELECT a seguir usa a função `get_dynamodb()` para recuperar o local para o ID do dispositivo especificado:

```
SELECT *, get_dynamodb("InServiceDevices", "deviceId", id,
"arn:aws:iam::12345678910:role/getdynamo").location AS location FROM 'some/
topic'
```

Note

- É possível chamar `get_dynamodb()` no máximo uma vez por instrução SQL. Chamar `get_dynamodb()` várias vezes em uma única instrução SQL faz com que a regra seja encerrada sem chamar nenhuma ação.
- Se `get_dynamodb()` retornar mais de 8 KB de dados, a ação da regra não poderá ser chamada.

get_secret (secretId, SecretType, chave, roleArn)

Recupera o valor do criptografado`SecretString`ou`SecretBinary`campo da versão atual de um segredo emAWS Secrets Manager. Para obter mais informações sobre como criar e manter segredos, consulteCreateSecret,UpdateSecret, ePutSecretValue.

`get_secret()` utiliza os seguintes parâmetros:

secretId

String: O nome de recurso da Amazon (ARN) ou o nome amigável do segredo a ser recuperado.

SecretType

String: O tipo de segredo. Valores válidos: `SecretString` | `SecretBinary`.

SecretString

- Para segredos que você cria como objetos JSON usando as APIs, oAWS CLI, ou oAWS Secrets ManagerConsole do :
  - Se você especificar um valor para a ação`key`, essa função retorna o valor da chave especificada.
  - Se você não especificar um valor para a ação`key`, esta função retorna todo o objeto JSON.
- Para segredos que você cria como objetos não JSON usando as APIs ou oAWS CLI:
  - Se você especificar um valor para a ação`key`parâmetro, essa função falha com uma exceção.
  - Se você não especificar um valor para a ação`key`, esta função retorna o conteúdo do segredo.

SecretBinary

- Se você especificar um valor para a ação`key`parâmetro, essa função falha com uma exceção.
- Se você não especificar um valor para a ação`key`, essa função retorna o valor secreto como uma string UTF-8 codificada em base64.

chave

(Opcional) String: O nome da chave dentro de um objeto JSON armazenado na`SecretString`campo de um segredo. Use esse valor quando quiser recuperar somente o valor de uma chave armazenada em um segredo em vez de todo o objeto JSON.

Se você especificar um valor para esse parâmetro e o segredo não contiver um objeto JSON dentro de seu`SecretString`campo, essa função falha com uma exceção.

roleArn

String: Um ARN de função do`secretsmanager:GetSecretValue`e`secretsmanager:DescribeSecret`Permissões

Note

Essa função sempre retorna a versão atual do segredo (a versão com a`AWSCURRENT`tag).
OAWS IoTO mecanismo de regras armazena em cache cada segredo por até 15 minutos. Como resultado, o mecanismo de regras pode levar até 15 minutos para atualizar um segredo. Isso significa que se você recuperar um segredo até 15 minutos após uma atualização comAWS Secrets Manager, essa função pode retornar a versão mais antiga.
Esta função não é medida e é gratuita para uso, masAWS Secrets ManagerHá cobranças aplicáveis. Por causa do mecanismo de cache secreto, o mecanismo de regras ocasionalmente chamaAWS Secrets Manager. Como o mecanismo de regras é um serviço totalmente distribuído, você pode ver várias chamadas de API do Secrets Manager a partir do mecanismo de regras durante a janela de armazenamento em cache de 15 minutos.

Exemplos:

Você pode usar o`get_secret`em um cabeçalho de autenticação em uma ação de regra HTTPS, como no exemplo de autenticação de chave de API a seguir.

```
"API_KEY": "${get_secret('API_KEY', 'SecretString', 'API_KEY_VALUE',
 'arn:aws:iam::12345678910:role/getsecret')}"
```

Para obter mais informações sobre a ação da regra HTTPS, consultethe section called “HTTP” (p. 504).

get_thing_shadow(thingName, shadowName, roleARN)

Retorna a sombra da coisa especificada. Compatível com a versão de 23/03/2016 do SQL e posteriores.

thingName

String: O nome da coisa cuja sombra você deseja recuperar.

shadowName

(Opcional) String: O nome do shadow. Esse parâmetro é necessário somente ao fazer referência a sombras nomeadas.

roleArn

String: Um ARN de função do`iot:GetThingShadow`permissão.

Exemplos:

Quando usado com uma sombra nomeada, forneça o parâmetro `shadowName`.

```
SELECT * from 'topic/subtopic'
WHERE
    get_thing_shadow("MyThing","MyThingShadow","arn:aws:iam::123456789012:role/
AllowsThingShadowAccess")
    .state.reported.alarm = 'ON'
```

Quando usado com uma sombra sem nome, omita o parâmetro `shadowName`.

```
SELECT * from 'topic/subtopic'
WHERE
    get_thing_shadow("MyThing","arn:aws:iam::123456789012:role/AllowsThingShadowAccess")
    .state.reported.alarm = 'ON'
```

Funções de hashing

A AWS IoT fornece as seguintes funções de hashing:

- md2
- md5
- sha1
- sha224
- sha256
- sha384
- sha512

Todas as funções de hash esperam um argumento de string. O resultado é o valor de hash da string em questão. Conversões de string padrão se aplicam a argumentos de não string. Todas as funções de hash são compatíveis com a versão de 08/10/2015 do SQL e posteriores.

Exemplos:

`md2("hello")` = "a9046c73e00331af68917d3804f70655"

`md5("hello")` = "5d41402abc4b2a76b9719d911017c592"

indexof(String, String)

Apresenta o primeiro índice (base em 0) do segundo argumento como uma substring no primeiro argumento. Os dois argumentos são esperados como strings. Os argumentos que não são strings estão sujeitos às regras padrão de conversão de string. Essa função não se aplica a matrizes, apenas a strings. Compatível com a versão de 23/03/2016 do SQL e posteriores.

Exemplos:

`indexof("abcd", "bc")` = 1

isNull()

Retorna verdadeiro se o argumento é o valor `Null`. Compatível com a versão de 08/10/2015 do SQL e posteriores.

Exemplos:

`isNull(5)` = false.

`isNull(Null)` = true.

Tipo de argumento	Result
`Int`	false
`Decimal`	false
`Boolean`	false
`String`	false
`Array`	false
`Object`	false
`Null`	true
`Undefined`	false

isUndefined()

Retorna verdadeiro se o argumento for `Undefined`. Compatível com a versão de 23/03/2016 do SQL e posteriores.

Exemplos:

`isUndefined(5)` = false.

`isUndefined(floor([1,2,3]))` = true.

Tipo de argumento	Result
`Int`	false
`Decimal`	false
`Boolean`	false
`String`	false
`Array`	false
`Object`	false
`Null`	false
`Undefined`	true

length(String)

Gera o número de caracteres na string fornecida. As regras padrão de conversão se aplicam a argumentos de não `String`. Compatível com a versão de 23/03/2016 do SQL e posteriores.

Exemplos:

`length("hi")` = 2

`length(false)` = 5

ln(Decimal)

Gera o logaritmo natural do argumento. Argumentos `Decimal` são arredondados para dobrar a precisão antes da aplicação da função. Compatível com a versão de 08/10/2015 do SQL e posteriores.

Exemplo: `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. Compatível com a versão de 08/10/2015 do SQL e posteriores.

Exemplo: `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`.

Tipo de argumento	Result
Não definido	`Undefined`.

lower(String)

Gera a versão em letra minúscula da `String` fornecida. Argumentos de não string são convertidos em strings usando as regras padrão de conversão. Compatível com a versão de 08/10/2015 do SQL e posteriores.

Exemplos:

`lower("HELLO")` = "hello".

`lower(["HELLO"])` = "[\"hello\"]".

lpad(String, Int)

Gera o argumento de `String`, preenchido no lado esquerdo com o número de espaços especificado pelo segundo argumento. O argumento `Int` deve estar entre 0 e 1000. Se o valor fornecido estiver fora desse intervalo válido, o argumento será definido para o valor válido mais próximo (0 ou 1.000). Compatível com a versão de 08/10/2015 do SQL e posteriores.

Exemplos:

`lpad("hello", 2)` = " hello".

`lpad(1, 3)` = " 1"

Tipo de argumento 1	Tipo de argumento 2	Result
`String`	`Int`	`String`, a `String` p com um número de e
`String`	`Decimal`	O argumento `Decima` próximo, e a `String` número especificado
`String`	`String`	O segundo argument que é arredondado pa será preenchida com à esquerda. Se o seg convertido em um `In`
Outros valores	`Int`/`Decimal`/`String`	O primeiro valor será as conversões padrão nessa `String`. Se nã resultado será `Undef`
Qualquer valor	Outros valores	`Undefined`.

ltrim(String)

Remove todos os espaços em branco iniciais (caracteres de tabulação e espaços) da `String` fornecida. Compatível com a versão de 08/10/2015 do SQL e posteriores.

Exemplo:

`Ltrim(" h i ")` = "hi ".

Tipo de argumento	Result
`Int`	A representação de `String` do `Int` com todos os espaços em branco iniciais removidos.
`Decimal`	A representação de `String` do `Decimal` com todos os espaços em branco iniciais removidos.
`Boolean`	A representação de `String` do booliano ("verdadeiro" ou "falso") com todos os espaços em branco iniciais removidos.
`String`	O argumento com todos os espaços em branco iniciais removidos.
Array	A representação de `String` de `Array` (usando as regras padrão de conversão) com todos os espaços em branco iniciais removidos.
Objeto	A representação de `String` do objeto (usando as regras padrão de conversão) com todos os espaços em branco iniciais removidos.
Nulo	`Undefined`.
Não definido	`Undefined`.

machinelearning_predict (modelId, roleArn, registro)

Usar a`machinelearning_predict`Para fazer previsões usando os dados de uma mensagem MQTT com base em um modelo Amazon Machine Learning (Amazon ML). Compatível com a versão de 08/10/2015 do SQL e posteriores. Os argumentos da função `machinelearning_predict` são:

modelId

O ID do modelo no qual executar a previsão. O endpoint em tempo real do modelo deve ser habilitado.

roleArn

A função do IAM que tem uma política com`machinelearning:Predict`e`machinelearning:GetMLModel`Permissões e permite acessar o modelo no qual a previsão é executada.

record

Os dados a serem passados na API Predict do Amazon ML. Isso deve ser representado como um único objeto JSON de camada. Se o registro for um objeto JSON de vários níveis, o registro será aplainado ao serializar seus valores. Por exemplo, o seguinte JSON:

```
{ "key1": {"innerKey1": "value1"}, "key2": 0}
```

se tornaria:

```
{ "key1": "{\"innerKey1\": \"value1\"}", "key2": 0}
```

A função gera um objeto JSON com os seguintes campos:

predictedLabel

A classificação da entrada com base no modelo.

detalhes

Contém os seguintes atributos:

PredictiveModelTipo

O tipo do modelo. Os valores válidos são REGRESSION, BINARY, MULTICLASS.

Algorithm

O algoritmo usado pelo Amazon ML para fazer previsões. O valor deve ser SGD.

predictedScores

Contém a pontuação de classificação bruta correspondente a cada rótulo.

predictedValue

O valor previsto pelo Amazon ML.

mod(Decimal, Decimal)

Gera o restante da divisão do primeiro argumento pelo segundo argumento. Equivale a remainder(Decimal, Decimal) (p. 610). Você também pode usar "%" como um operador infixo para a mesma funcionalidade do módulo. Compatível com a versão de 08/10/2015 do SQL e posteriores.

Exemplo: `mod(8, 3)` = 2.

Operando esquerdo	Operando direito	Resultado
`Int`	`Int`	`Int`, o primeiro argun argumento.
`Int/Decimal`	`Int/Decimal`	`Decimal`, o primeiro operando.
`String/Int/Decimal`	`String/Int/Decimal`	Se todas as strings fo resultado será o prime segundo argumento.
Outros valores	Outros valores	`Undefined`.

nanvl(AnyValue, AnyValue)

Gera o primeiro argumento se for um `Decimal` válido. Caso contrário, o segundo argumento é retornado. Compatível com a versão de 08/10/2015 do SQL e posteriores.

Exemplo: `Nanvl(8, 3)` = 8.

Tipo de argumento 1	Tipo de argumento 2	Resultado
Não definido	Qualquer valor	O segundo argumento
Nulo	Qualquer valor	O segundo argumento

Tipo de argumento 1	Tipo de argumento 2	Resultado
`Decimal` (NaN)	Qualquer valor	O segundo argumento
`Decimal` (não NaN)	Qualquer valor	O primeiro argumento
Outros valores	Qualquer valor	O primeiro argumento

newuuid()

Gera uma UUID aleatória de 16 bytes. Compatível com a versão de 08/10/2015 do SQL e posteriores.

Exemplo: `newuuid()` = `123a4567-b89c-12d3-e456-789012345000`

numbytes(String)

Gera o número de bytes na codificação UTF-8 da string fornecida. As regras padrão de conversão se aplicam a argumentos de não `String`. Compatível com a versão de 23/03/2016 do SQL e posteriores.

Exemplos:

`numbytes("hi")` = 2

`numbytes("€")` = 3

parse_time (String, Long [, String])

Use a função `parse_time` para formatar um timestamp em um formato legível de data/hora. Compatível com a versão de 23/03/2016 do SQL e posteriores. Para converter uma string de carimbo de data/hora em milissegundos, consultetime_to_epoch (String, String) (p. 618).

O`parse_time`A função espera os seguintes argumentos:

pattern

(String) Um padrão de data/hora que segue aISO 8601formato padrão. Especificamente, a função suportaFormatos Joda-Time.

timestamp

(Long) O tempo a ser formatado em milissegundos desde o epoch do Unix. Consulte a função timestamp() (p. 619).

timezone

(String) O fuso horário da data/hora formatada. O padrão é "UTC". A função oferece suporte para fusos horários Joda-Time. Esse argumento é opcional.

Exemplos:

Quando essa mensagem for publicada no tópico "A/B", a carga útil `{"ts": "1970.01.01 AD at 21:46:40 CST"}` será enviada para o bucket do S3:

```
{
    "ruleArn": "arn:aws:iot:us-east-2:ACCOUNT_ID:rule/RULE_NAME",
    "topicRulePayload": {
        "sql": "SELECT parse_time(\"yyyy.MM.dd G 'at' HH:mm:ss z\", 100000000, 'America/
Belize' ) as ts FROM 'A/B'",

        "ruleDisabled": false,
```

```
        "awsIotSqlVersion": "2016-03-23",
        "actions": [
            {
                "s3": {
                    "roleArn": "arn:aws:iam::ACCOUNT_ID:rule:role/ROLE_NAME",
                    "bucketName": "BUCKET_NAME",
                    "key": "KEY_NAME"
                }
            }
        ],
        "ruleName": "RULE_NAME"
    }
}
```

Quando essa mensagem for publicada no tópico 'A/B', uma carga útil semelhante a `{"ts": "2017.06.09 AD at 17:19:46 UTC"}` (mas com data/hora atual) será enviada para o bucket S3:

```
{
    "ruleArn": "arn:aws:iot:us-east-2:ACCOUNT_ID:rule/RULE_NAME",
    "topicRulePayload": {
        "sql": "SELECT parse_time(\"yyyy.MM.dd G 'at' HH:mm:ss z\", timestamp() ) as ts
 FROM 'A/B'",
        "awsIotSqlVersion": "2016-03-23",
        "ruleDisabled": false,
        "actions": [
            {
                "s3": {
                    "roleArn": "arn:aws:iam::ACCOUNT_ID:rule:role/ROLE_NAME",
                    "bucketName": "BUCKET_NAME",
                    "key": "KEY_NAME"
                }
            }
        ],
        "ruleName": "RULE_NAME"
    }
}
```

`parse_time()` também pode ser usado como um modelo de substituição. Por exemplo, quando esta mensagem for publicada no tópico "A/B", a carga útil será enviada para o bucket S3 com chave = "2017":

```
{
    "ruleArn": "arn:aws:iot:us-east-2:ACCOUNT_ID:rule/RULE_NAME",
    "topicRulePayload": {
        "sql": "SELECT * FROM 'A/B'",
        "awsIotSqlVersion": "2016-03-23",
        "ruleDisabled": false,
        "actions": [{
            "s3": {
                "roleArn": "arn:aws:iam::ACCOUNT_ID:rule:role/ROLE_NAME",
                "bucketName": "BUCKET_NAME",
                "key": "${parse_time('yyyy', timestamp(), 'UTC')}"
            }
        }],
        "ruleName": "RULE_NAME"
    }
}
```

power(Decimal, Decimal)

Gera o primeiro argumento elevado ao segundo argumento. Argumentos `Decimal` são arredondados para dobrar a precisão antes da aplicação da função. Compatível com a versão de 08/10/2015 do SQL e posteriores. Compatível com a versão de 08/10/2015 do SQL e posteriores.

Exemplo: `power(2, 5)` = 32,0.

Tipo de argumento 1	Tipo de argumento 2	Resultado
`Int/Decimal`	`Int/Decimal`	Um `Decimal` (com pr elevado para o poder
`Int/Decimal/String`	`Int/Decimal/String`	Um `Decimal` (com pr elevado para o poder strings são convertida alguma `String` em `D` `Undefined`.
Outros valores	Outros valores	`Undefined`.

principal()

Retorna o principal usado pelo dispositivo para autenticação, com base em como a mensagem de acionamento foi publicada. A tabela a seguir descreve a entidade principal retornada para cada protocolo e método de publicação.

Como a mensagem é publicada	Protocolo	Tipo de credencial
Cliente MQTT	MQTT	Certificado de disposi
Cliente MQTT do console do AWS IoT	MQTT	Usuário ou função do
AWS CLI	HTTP	Usuário ou função do
SDK de dispositivo do AWS IoT	MQTT	Certificado de disposi
SDK de dispositivo do AWS IoT	MQTT pelo WebSocket	Usuário ou função do

Os exemplos a seguir mostram os diferentes tipos de valores que`principal()`pode retornar:

- Thumbprint do certificado X.509: `ba67293af50bf2506f5f93469686da660c7c844e7b3950bfb16813e0d31e9373`
- ID da função do IAM e nome da sessão:`ABCD1EFG3HIJK2LMNOP5:my-session-name`
- retorna um ID de usuário:`ABCD1EFG3HIJK2LMNOP5`

rand()

Gera um pseudo-aleatório, uniformemente distribuído entre 0,0 e 1,0. Compatível com a versão de 08/10/2015 do SQL e posteriores.

Exemplo:

`rand()` = 0,8231909191640703

regexp_matches(String, String)

Retorna verdadeiro se a string (primeiro argumento) contiver uma correspondência para a expressão regular (segundo argumento).

Exemplo:

`regexp_matches("aaaa", "a{2,}")` = true.

`regexp_matches("aaaa", "b")` = false.

Primeiro argumento:

Tipo de argumento	Result
`Int`	A representação da `String` do `Int`.
`Decimal`	A representação da `String` do `Decimal`.
`Boolean`	A representação da `String` do booleanoo ("verdadeiro" ou "falso").
`String`	A ferramenta `String`.
Array	A representação da `String` da `Array` (usando as regras padrão de conversão).
Objeto	A representação da `String` do Objeto (usando as regras padrão de conversão).
Nulo	`Undefined`.
Não definido	`Undefined`.

Segundo argumento:

Deve ser uma expressão regex válida. Tipos de não string são convertidos em `String` usando as regras padrão de conversão. Dependendo do tipo, a string resultante pode não ser uma expressão regular válida. Se o argumento (convertido) não for uma regex válida, o resultado será `Undefined`.

regexp_replace(String, String, String)

Substitui todas as ocorrências do segundo argumento (expressão regular) no primeiro argumento com o terceiro argumento. Faz referência a grupos de captura com "$". Compatível com a versão de 08/10/2015 do SQL e posteriores.

Exemplo:

`regexp_replace("abcd", "bc", "x")` = "axd".

`regexp_replace("abcd", "b(.*)d", "$1")` = "ac".

Primeiro argumento:

Tipo de argumento	Result
`Int`	A representação da `String` do `Int`.
`Decimal`	A representação da `String` do `Decimal`.
`Boolean`	A representação da `String` do booleanoo ("verdadeiro" ou "falso").
`String`	O valor de origem.

Tipo de argumento	Result
Array	A representação da `String` da `Array` (usando as regras padrão de conversão).
Objeto	A representação da `String` do Objeto (usando as regras padrão de conversão).
Nulo	`Undefined`.
Não definido	`Undefined`.

Segundo argumento:

Deve ser uma expressão regex válida. Tipos de não string são convertidos em `String` usando as regras padrão de conversão. Dependendo do tipo, a string resultante pode não ser uma expressão regular válida. Se o argumento (convertido) não for uma expressão regex válida, o resultado será `Undefined`.

Terceiro argumento:

Deve ser uma string de substituição regex válida. (Pode fazer referência a grupos de captura.) Tipos de não string são convertidos em `String` usando as regras padrão de conversão. Se o argumento (convertido) não for uma string de substituição regex válida, o resultado será `Undefined`.

regexp_substr(String, String)

Localiza a primeira correspondência do segundo parâmetro (regex) no primeiro parâmetro. Faz referência a grupos de captura com "$". Compatível com a versão de 08/10/2015 do SQL e posteriores.

Exemplo:

`regexp_substr("hihihello", "hi")` = "hi"

`regexp_substr("hihihello", "(hi)*")` = "hihi"

Primeiro argumento:

Tipo de argumento	Result
`Int`	A representação da `String` do `Int`.
`Decimal`	A representação da `String` do `Decimal`.
`Boolean`	A representação da `String` do booleanoo ("verdadeiro" ou "falso").
`String`	O argumento da `String`.
Array	A representação da `String` da `Array` (usando as regras padrão de conversão).
Objeto	A representação da `String` do Objeto (usando as regras padrão de conversão).
Nulo	`Undefined`.
Não definido	`Undefined`.

Segundo argumento:

Deve ser uma expressão regex válida. Tipos de não string são convertidos em `String` usando as regras padrão de conversão. Dependendo do tipo, a string resultante pode não ser uma expressão regular válida. Se o argumento (convertido) não for uma expressão regex válida, o resultado será `Undefined`.

remainder(Decimal, Decimal)

Gera o restante da divisão do primeiro argumento pelo segundo argumento. Equivale a mod(Decimal, Decimal) (p. 604). Você também pode usar "%" como um operador infixo para a mesma funcionalidade do módulo. Compatível com a versão de 08/10/2015 do SQL e posteriores.

Exemplo: `remainder(8, 3)` = 2.

Operando esquerdo	Operando direito	Resultado
`Int`	`Int`	`Int`, o primeiro argun argumento.
`Int/Decimal`	`Int/Decimal`	`Decimal`, o primeiro operando.
`String/Int/Decimal`	`String/Int/Decimal`	Se todas as strings fo resultado será o prime segundo argumento.
Outros valores	Outros valores	`Undefined`.

replace(String, String, String)

Substitui todas as ocorrências do segundo argumento no primeiro argumento com o terceiro argumento. Compatível com a versão de 08/10/2015 do SQL e posteriores.

Exemplo:

`replace("abcd", "bc", "x")` = `"axd"`.

`replace("abcdabcd", "b", "x")` = `"axcdaxcd"`.

Todos os argumentos

Tipo de argumento	Result
`Int`	A representação da `String` do `Int`.
`Decimal`	A representação da `String` do `Decimal`.
`Boolean`	A representação da `String` do booleanoo ("verdadeiro" ou "falso").
`String`	O valor de origem.
Array	A representação da `String` da `Array` (usando as regras padrão de conversão).
Objeto	A representação da `String` do Objeto (usando as regras padrão de conversão).
Nulo	`Undefined`.
Não definido	`Undefined`.

rpad(String, Int)

Gera o argumento da string, preenchido no lado direito com o número de espaços especificado no segundo argumento. O argumento `Int` deve estar entre 0 e 1000. Se o valor fornecido estiver fora desse intervalo válido, o argumento será definido para o valor válido mais próximo (0 ou 1.000). Compatível com a versão de 08/10/2015 do SQL e posteriores.

Exemplos:

`rpad("hello", 2)` = "hello ".

`rpad(1, 3)` = "1 ".

Tipo de argumento 1	Tipo de argumento 2	Result
`String`	`Int`	A `String` é preenchida no lado direito com um número de espaços igual ao `Int` fornecido.
`String`	`Decimal`	O argumento `Decimal` será arredondado para o `Int` mais próximo, e a string será preenchida no lado direito com um número de espaços igual ao `Int` fornecido.
`String`	`String`	O segundo

Tipo de argumento 1	Tipo de argumento 2	Result
		argumento é convertido em um `Decimal`, que é arredondado para o `Int` mais próximo. A `String` é preenchida no lado direito com um número de espaços igual ao valor `Int`.

Tipo de argumento 1	Tipo de argumento 2	Result
Outros valores	`Int`/`Decimal`/`String`	O primeiro valor será convertido em uma `String` usando as conversões padrão, e a função rpad será aplicada nessa `String`. Se não for possível fazer a conversão, o resultado será `Undefined`.
Qualquer valor	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. Compatível com a versão de 08/10/2015 do SQL e posteriores.

Exemplo: `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.

Tipo de argumento	Result
`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`.

rtrim(String)

Remove todos os espaços em branco finais (caracteres de tabulação e espaços) da `String` fornecida. Compatível com a versão de 08/10/2015 do SQL e posteriores.

Exemplos:

`rtrim(" h i ")` = " h i"

Tipo de argumento	Result
`Int`	A representação da `String` do `Int`.
`Decimal`	A representação da `String` do `Decimal`.
`Boolean`	A representação da `String` do booleanoo ("verdadeiro" ou "falso").
Array	A representação da `String` da `Array` (usando as regras padrão de conversão).
Objeto	A representação da `String` do Objeto (usando as regras padrão de conversão).
Nulo	`Undefined`.
Não definido	`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. Compatível com a versão de 08/10/2015 do SQL e posteriores.

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`.

Tipo de argumento	Result
`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`. Compatível com a versão de 08/10/2015 do SQL e posteriores.
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. Compatível com a versão de 08/10/2015 do SQL e posteriores.

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

Tipo de argumento	Result
`Int`	`Decimal` (com precisão dupla), o seno do argumento.
`Decimal`	`Decimal` (com precisão dupla), o seno do argumento.
`Boolean`	`Undefined`.
`String`	`Decimal` (com precisão dupla), o seno do argumento. Se a string não puder ser convertida em um `Decimal`, o resultado será `Undefined`.
Array	`Undefined`.
Objeto	`Undefined`.
Nulo	`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. Compatível com a versão de 08/10/2015 do SQL e posteriores.

Exemplo: `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`.

Tipo de argumento	Result
`String`	`Decimal` (com precisão dupla), o seno hiperbólico 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`.

substring (String, Int [, Int])

Espera uma `String` seguida de um ou dois valores `Int`. No caso de uma `String` e um único argumento `Int`, essa função gera a substring da `String` fornecida do índice `Int` fornecido (baseado em 0, inclusive) ao final da `String`. No caso de uma `String` e dois argumentos `Int`, essa função gera a substring da `String` fornecida no primeiro argumento de índice`Int` (baseado em 0, inclusive) para o segundo argumento de índice `Int` (baseado em 0, inclusive). Índices inferiores a zero são definidos como zero. Índices superiores ao comprimento da `String` são definidos para o comprimento da `String`. No caso da versão de argumento três, se o primeiro índice for superior ou igual ao segundo índice, o resultado será a `String` vazia.

Se os argumentos fornecidos não forem (*`String`*, *`Int`*) nem (*`String`*, *`Int`*, *`Int`*), as conversões padrão serão aplicadas aos argumentos na tentativa de convertê-los nos tipos corretos. Se os tipos não puderem ser convertidos, o resultado da função será `Undefined`. Compatível com a versão de 08/10/2015 do SQL e posteriores.

Exemplos:

`substring("012345", 0)` = "012345".

`substring("012345", 2)` = "2345".

`substring("012345", 2.745)` = "2345".

`substring(123, 2)` = "3".

`substring("012345", -1)` = "012345".

`substring(true, 1.2)` = "rue".

`substring(false, -2.411E247)` = "false".

`substring("012345", 1, 3)` = "12".

`substring("012345", -50, 50)` = "012345".

`substring("012345", 3, 1)` = "".

sql_version()

Retorna a versão SQL especificada nesta regra. Compatível com a versão de 08/10/2015 do SQL e posteriores.

Exemplo:

`sql_version()` = "23/03/2016"

sqrt(Decimal)

Gera a raiz quadrada de um número. Argumentos `Decimal` são arredondados para dobrar a precisão antes da aplicação da função. Compatível com a versão de 08/10/2015 do SQL e posteriores.

Exemplo: `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`.
Objeto	`Undefined`.
Nulo	`Undefined`.
Não definido	`Undefined`.

startswith(String, String)

Gera `Boolean`, independentemente de o primeiro argumento da string começar com o segundo argumento da string. Se um dos argumentos for `Null` ou `Undefined`, o resultado será `Undefined`. Compatível com a versão de 08/10/2015 do SQL e posteriores.

Exemplo:

`startswith("ranger","ran")` = true

Tipo de argumento 1	Tipo de argumento 2	Result
`String`	`String`	Independentemente d segunda string.
Outros valores	Outros valores	Os dois argumentos s as regras padrão de c independentemente d a segunda string. Se `Undefined`, o resulta

tan(Decimal)

Gera a tangente de um número em radianos. Valores `Decimal` são arredondados para dobrar a precisão antes da aplicação da função. Compatível com a versão de 08/10/2015 do SQL e posteriores.

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`.
Objeto	`Undefined`.
Nulo	`Undefined`.
Não definido	`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. Compatível com a versão de 08/10/2015 do SQL e posteriores.

Exemplo: `tanh(2.3)` = 0,9800963961914

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`.
Objeto	`Undefined`.
Nulo	`Undefined`.
Não definido	`Undefined`.

time_to_epoch (String, String)

Usar a`time_to_epoch`para converter uma string de carimbo de data/hora em um número de milissegundos no tempo de época Unix. Compatível com a versão de 23/03/2016 do SQL e posteriores. Para converter milissegundos em uma string de carimbo de data/hora formatada, consulteparse_time (String, Long [, String]) (p. 605).

O`time_to_epoch`A função espera os seguintes argumentos:

timestamp

(String) A string de carimbo de data/hora a ser convertida em milissegundos desde a época do Unix. Se a string de carimbo de data/hora não especificar um fuso horário, a função usará o fuso horário UTC.

pattern

(String) Um padrão de data/hora que segue aISO 8601formato padrão. Especificamente, a função suportaFormatos de hora JDK11.

Exemplos:

`time_to_epoch("2020-04-03 09:45:18 UTC+01:00", "yyyy-MM-dd HH:mm:ss VV")`= 1585903518000

`time_to_epoch("18 December 2015", "dd MMMM yyyy")`= 1450396800000

`time_to_epoch("2007-12-03 10:15:30.592 America/Los_Angeles", "yyyy-MM-dd HH:mm:ss.SSS z")`= 1196705730592

timestamp()

Gera o timestamp atual em milissegundos de 00:00:00 UTC (Tempo Universal Coordenado), quinta-feira, 1° de janeiro de 1970, como observado pelo mecanismo de regras da AWS IoT. Compatível com a versão de 08/10/2015 do SQL e posteriores.

Exemplo: `timestamp()` = `1481825251155`

topic(Decimal)

Gera o tópico para o qual a mensagem que acionou a regra foi enviada. Se nenhum parâmetro for especificado, todo o tópico será gerado. O parâmetro `Decimal` é usado para especificar um segmento de tópico específico, com 1 designando o primeiro segmento. Para o tópico `foo/bar/baz`, topic(1) retornará `foo`, topic(2) retornará `bar`, e assim por diante. Compatível com a versão de 08/10/2015 do SQL e posteriores.

Exemplos:

`topic()` = "things/myThings/thingOne"

`topic(1)` = "coisas"

Quando a Ingestão básica (p. 567) for usada, o prefixo inicial do tópico (`$aws/rules/`*`rule-name`*) não estará disponível na função topic(). Por exemplo, considerando o tópico:

`$aws/rules/BuildingManager/Buildings/Building5/Floor2/Room201/Lights`

`topic()` = "Buildings/Building5/Floor2/Room201/Lights"

`topic(3)` = "Floor2"

traceid()

Gera o ID de rastreamento (UUID) da mensagem MQTT ou `Undefined` caso a mensagem não tenha sido enviada por MQTT. Compatível com a versão de 08/10/2015 do SQL e posteriores.

Exemplo:

`traceid()` = "12345678-1234-1234-1234-123456789012"

transform (String, Objeto, Array)

Retorna uma matriz de objetos que contém o resultado da transformação especificada do`Object`parâmetro no`Array`parâmetro .

Compatível com a versão de 23/03/2016 do SQL e posteriores.

String

O modo de transformação a ser usado. Consulte a tabela a seguir para ver os modos de transformação suportados e como eles criam o`Result`do`Object`e`Array`parâmetros.

Objeto

Um objeto que contém os atributos a serem aplicados a cada elemento do`Array`.

Array

Uma matriz de objetos no qual os atributos do`Object`são aplicados.

Cada objeto nesta matriz corresponde a um objeto na resposta da função. Cada objeto na resposta da função contém os atributos presentes no objeto original e os atributos fornecidos por`Object`conforme determinado pelo modo de transformação especificado em`String`.

Parâmetro **`String`**	Parâmetro **`Object`**	Parâmetro **`Array`**	Result
`enrichArray`	Objeto	Matriz de objetos	Uma matriz de objetos na qual cada objeto contém os atributos de um elemento do`Array`parâmetro e os atributos do`Object`parâmetro .
Qualquer outro valor	Qualquer valor	Qualquer valor	Não definido

Note

A matriz retornada por esta função é limitada a 128 KiB.

Exemplo 1 da função Transform

Este exemplo mostra como a ação`transform()`produz uma única matriz de objetos de um objeto de dados e uma matriz.

Neste exemplo, a seguinte mensagem é publicada no tópico MQTT`A/B`.

```
{
    "attributes": {
        "data1": 1,
        "data2": 2
    },
    "values": [
        {
            "a": 3
        },
        {
            "b": 4
        },
```

```
        {
            "c": 5
        }
    ]
}
```

Esta instrução SQL para uma ação de regra de tópico usa otransform()função com um`String`valor de`enrichArray`. Neste exemplo,`Object`é o`attributes`propriedade da carga útil da mensagem e`Array`é o`values`array, que contém três objetos.

```
select value transform("enrichArray", attributes, values) from 'A/B'
```

Ao receber a carga útil da mensagem, a instrução SQL é avaliada para a resposta a seguir.

```
[
  {
    "a": 3,
    "data1": 1,
    "data2": 2
  },
  {
    "b": 4,
    "data1": 1,
    "data2": 2
  },
  {
    "c": 5,
    "data1": 1,
    "data2": 2
  }
]
```

Função Transform exemplo 2

Este exemplo mostra como a ação`transform()`função pode usar valores literais para incluir e renomear atributos individuais a partir da carga útil da mensagem.

Neste exemplo, a seguinte mensagem é publicada no tópico MQTT`A/B`. Essa é a mesma mensagem que foi usada emthe section called “Exemplo 1 da função Transform” (p. 620).

```
{
    "attributes": {
        "data1": 1,
        "data2": 2
    },
    "values": [
        {
            "a": 3
        },
        {
            "b": 4
        },
        {
            "c": 5
        }
    ]
}
```

Esta instrução SQL para uma ação de regra de tópico usa otransform()função com um`String`valor de`enrichArray`. O`Object`no`transform()`função tem um único atributo chamado`key`com o valor

de`attributes.data1`na carga útil da mensagem e`Array`é o`values`, que contém os mesmos três objetos usados no exemplo anterior.

```
select value transform("enrichArray", {"key": attributes.data1}, values) from 'A/B'
```

Ao receber a carga útil da mensagem, esta instrução SQL é avaliada para a resposta a seguir. Observe como o`data1`propriedade é nomeada`key`na resposta.

```
[
  {
    "a": 3,
    "key": 1
  },
  {
    "b": 4,
    "key": 1
  },
  {
    "c": 5,
    "key": 1
  }
]
```

Função Transform exemplo 3

Este exemplo mostra como a ação`transform()`função pode ser usada em cláusulas SELECT aninhadas para selecionar vários atributos e criar novos objetos para processamento subsequente.

Neste exemplo, a seguinte mensagem é publicada no tópico MQTT`A/B`.

```
{
  "data1": "example",
  "data2": {
    "a": "first attribute",
    "b": "second attribute",
    "c": [
      {
        "x": {
          "someInt": 5,
          "someString": "hello"
        },
        "y": true
      },
      {
        "x": {
          "someInt": 10,
          "someString": "world"
        },
        "y": false
      }
    ]
  }
}
```

O`Object`para esta função de transformação é o objeto retornado pela instrução SELECT, que contém o`a`e`b`elementos da mensagem`data2`objeto. O`Array`O parâmetro consiste nos dois objetos do`data2.c`Matriz na mensagem original.

```
select value transform('enrichArray', (select a, b from data2), (select value c from
 data2)) from 'A/B'
```

Com a mensagem anterior, a instrução SQL é avaliada para a resposta a seguir.

```
[
  {
    "x": {
      "someInt": 5,
      "someString": "hello"
    },
    "y": true,
    "a": "first attribute",
    "b": "second attribute"
  },
  {
    "x": {
      "someInt": 10,
      "someString": "world"
    },
    "y": false,
    "a": "first attribute",
    "b": "second attribute"
  }
]
```

O array retornado nesta resposta pode ser usado com ações de regras de tópico que suportam`batchMode`.

trim(String)

Remove todos os espaços em branco iniciais e finais da `String` fornecida. Compatível com a versão de 08/10/2015 do SQL e posteriores.

Exemplo:

`Trim(" hi ")` = "hi"

Tipo de argumento	Result
`Int`	A representação de `String` de `Int` com todos os espaços em branco iniciais e finais removidos.
`Decimal`	A representação de `String` de `Decimal` com todos os espaços em branco iniciais e finais removidos.
`Boolean`	A representação de `String` de `Boolean` ("verdadeiro" ou "falso") com todos os espaços em branco iniciais e finais removidos.
`String`	A `String` com todos os espaços em branco iniciais e finais removidos.
Array	A representação da `String` da `Array` usando as regras padrão de conversão.
Objeto	A representação da `String` do Objeto usando as regras padrão de conversão.
Nulo	`Undefined`.
Não definido	`Undefined`.

trunc(Decimal, Int)

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. Compatível com a versão de 08/10/2015 do SQL e posteriores.

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`	`Int/Decimal`	O primeiro argumento descrito pelo segundo se não for um `Int`, se próximo.
`Int/Decimal/String`	`Int/Decimal`	O primeiro argumento descrito pelo segundo se não for um `Int`, se próximo. A `String` é Se não for possível co `Undefined`.
Outros valores		`Undefined`.

upper(String)

Gera a versão em letra maiúscula da `String` fornecida. Os argumentos de não`String` são convertidos em `String` usando as regras de conversão padrão. Compatível com a versão de 08/10/2015 do SQL e posteriores.

Exemplos:

`upper("hello")` = "HELLO"

`upper(["hello"])` = "[\"HELLO\"]"

Literais

Você pode especificar diretamente objetos literais nas cláusulas WHERE e SELECT da regra SQL, que pode ser útil para transmitir informações.

> **Note**
>
> Literais estão disponíveis somente ao usar a versão de 23/03/2016 do SQL ou posteriores.

A sintaxe de objeto JSON é usada (pares de chave-valor, separado por vírgula, em que as chaves são strings, e os valores são valores JSON, entre chaves {}). Por exemplo:

Carga útil de entrada publicada no tópico `topic/subtopic`: `{"lat_long": [47.606,-122.332]}`

Declaração do SQL: `SELECT {'latitude': get(lat_long, 0),'longitude':get(lat_long, 1)} as lat_long FROM 'topic/subtopic'`

A carga útil de saída resultante seria: `{"lat_long": {"latitude":47.606,"longitude":-122.332}}`.

Você também pode especificar diretamente matrizes nas cláusulas WHERE e SELECT da regra SQL, o que permite agrupar informações. A sintaxe JSON é usada (encapsular itens separados por vírgula entre colchetes [] para criar uma matriz literal). Por exemplo:

Carga útil de entrada publicada no tópico `topic/subtopic`: `{"lat": 47.696, "long": -122.332}`

Declaração do SQL: `SELECT [lat,long] as lat_long FROM 'topic/subtopic'`

A carga útil de saída resultante seria: `{"lat_long": [47.606,-122.332]}`.

Declarações de caso

As declarações de caso podem ser usadas para execução de ramificação, como uma declaração de troca.

Sintaxe:

```
CASE v WHEN t[1] THEN r[1]
  WHEN t[2] THEN r[2] ...
  WHEN t[n] THEN r[n]
  ELSE r[e] END
```

A expressão*v*é avaliado e correspondido para igualdade em relação ao*t[i]*valor de cada`WHEN`Cláusula. Se uma correspondência for encontrada, o correspondente*r[i]*expressão se torna o resultado do`CASE`instrução. O`WHEN`cláusulas são avaliadas em ordem para que, se houver mais de uma cláusula de correspondência, o resultado da primeira cláusula de correspondência se torne o resultado da`CASE`instrução. Se não houver correspondências,*r[e]*do`ELSE`cláusula é o resultado. Se não houver correspondência e não`ELSE`cláusula, o resultado é`Undefined`.

`CASE`As declarações exigem, pelo menos, um`WHEN`Cláusula. Uma`ELSE`cláusula é opcional.

Por exemplo:

Carga útil de entrada publicada no tópico`topic/subtopic`:

```
{
    "color":"yellow"
}
```

Declaração do SQL:

```
SELECT CASE color
        WHEN 'green' THEN 'go'
        WHEN 'yellow' THEN 'caution'
        WHEN 'red' THEN 'stop'
        ELSE 'you are not at a stop light' END as instructions
    FROM 'topic/subtopic'
```

A carga útil de saída resultante seria:

```
{
    "instructions":"caution"
}
```

> **Note**
>
> Se `v`é`Undefined`, o resultado da declaração do caso é`Undefined`.

Extensões JSON

Você pode usar as seguintes extensões para a sintaxe SQL padrão ANSI para facilitar o trabalho com objetos JSON aninhados.

"." Operador

Esse operador acessa membros em objetos e funções JSON incorporados identicamente a SQL padrão ANSI e JavaScript. Por exemplo:

```
SELECT foo.bar AS bar.baz FROM 'topic/subtopic'
```

seleciona o valor do`bar`propriedade no`foo`objeto da seguinte carga de mensagem enviada para o`topic/subtopic`tópico.

```
{
  "foo": {
    "bar": "RED",
    "bar1": "GREEN",
    "bar2": "BLUE"
  }
}
```

Se um nome de propriedade JSON incluir um caractere de hífen ou caracteres numéricos, a notação “ponto” simples não funcionará. Em vez disso, você deverá usar a açãoobter função (p. 595)para extrair o valor da propriedade.

Neste exemplo, a seguinte mensagem é enviada para a`iot/rules`tópico.

```
{
  "mydata": {
    "item2": {
      "0": {
        "my-key": "myValue"
      }
    }
  }
}
```

Normalmente, o valor de`my-key`seria identificado como nesta consulta.

```
SELECT * from iot/rules WHERE mydata.item2.0.my-key= "myValue"
```

No entanto, porque o nome da propriedade`my-key`contém um hífen e`item2`contém um caractere numérico, oobter função (p. 595)deve ser usado como mostra a consulta a seguir.

```
SELECT * from 'iot/rules' WHERE get(get(get(mydata,"item2"),"0"),"my-key") = "myValue"
```

`*`Operador

Funciona da mesma maneira que o caractere curinga `*` em SQL padrão ANSI. É usado somente na cláusula SELECT e cria outro objeto JSON contendo os dados da mensagem. Se a carga útil da mensagem não estiver no formato JSON, `*` gerará toda a carga útil da mensagem como bytes brutos. Por exemplo:

```
SELECT * FROM 'topic/subtopic'
```

Como aplicar uma função a um valor de atributo

Veja a seguir um exemplo de carga útil JSON que pode ser publicada por um dispositivo:

```
{
    "deviceid" : "iot123",
    "temp" : 54.98,
    "humidity" : 32.43,
    "coords" : {
        "latitude" : 47.615694,
        "longitude" : -122.3359976
    }
}
```

O exemplo a seguir aplica uma função a um valor de atributo em uma carga útil JSON:

```
SELECT temp, md5(deviceid) AS hashed_id FROM topic/#
```

O resultado dessa consulta é o seguinte objeto JSON:

```
{
   "temp": 54.98,
   "hashed_id": "e37f81fb397e595c4aeb5645b8cbbbd1"
}
```

Modelos de substituição

Você pode usar um modelo de substituição para aumentar os dados JSON gerados quando uma regra é acionada e a AWS IoT realiza uma ação. A sintaxe de um modelo de substituição é `${expressão}`, em que expressão pode ser qualquer expressão compatível com o AWS IoT na cláusula SELECT, na cláusula WHERE e em Ações de regra do AWS IoT (p. 485). Essa expressão pode ser conectada a um campo de ação em uma regra, permitindo que você configure dinamicamente uma ação. Na verdade, esse recurso substitui uma informação em uma ação. Isso inclui funções, operadores e informações presentes na carga da mensagem original.

> **Important**
>
> Como uma expressão em um modelo de substituição é avaliada separadamente da instrução "SELECT...", você não pode fazer referência a um alias criado com a cláusula AS. É possível fazer referência somente às informações presentes na carga original,funções (p. 581), eoperadores (p. 575).

Para obter mais informações sobre as expressões compatíveis, consulte Referência SQL do AWS IoT (p. 568).

As ações de regra a seguir oferecem suporte a modelos de substituição. Cada ação oferece suporte a campos diferentes que podem ser substituídos.

- Apache Kafka (p. 487)
- Alarmes do CloudWatch (p. 495)
- CloudWatch Logs (p. 496)
- Métricas do CloudWatch (p. 497)
- DynamoDB (p. 499)
- DynamoDBv2 (p. 501)
- Elasticsearch (p. 503)
- HTTP (p. 504)
- IoT Analytics (p. 531)
- IoT Events (p. 533)
- IoT SiteWise (p. 534)
- Kinesis Data Streams (p. 540)
- Kinesis Data Firehose (p. 538)
- Lambda (p. 542)
- OpenSearch (p. 544)
- Republish (p. 546)
- S3 (p. 547)
- SNS (p. 549)
- SQS (p. 551)
- Step Functions (p. 553)
- Timestream (p. 554)

Os modelos de substituição aparecem nos parâmetros de ação dentro de uma regra:

```
{
    "sql": "SELECT *, timestamp() AS timestamp FROM 'my/iot/topic'",
    "ruleDisabled": false,
    "actions": [{
        "republish": {
            "topic": "${topic()}/republish",
            "roleArn": "arn:aws:iam::123456789012:role/my-iot-role"
        }
    }]
}
```

Se essa regra for acionada pelo JSON a seguir publicado em `my/iot/topic`:

```
{
    "deviceid": "iot123",
    "temp": 54.98,
    "humidity": 32.43,
    "coords": {
        "latitude": 47.615694,
        "longitude": -122.3359976
    }
}
```

Depois, esta regra publica o JSON a seguir como `my/iot/topic/republish`, que o AWS IoT substitui de `${topic()}/republish`:

```
{
    "deviceid": "iot123",
    "temp": 54.98,
    "humidity": 32.43,
    "coords": {
        "latitude": 47.615694,
        "longitude": -122.3359976
    },
    "timestamp": 1579637878451
}
```

Consultas de objeto aninhado

Você pode usar cláusulas SELECT aninhadas para consultar atributos dentro de matrizes e objetos JSON internos. Compatível com a versão de 23/03/2016 do SQL e posteriores.

Considere a seguinte mensagem MQTT:

```
{
    "e": [
        { "n": "temperature", "u": "Cel", "t": 1234, "v": 22.5 },
        { "n": "light", "u": "lm", "t": 1235, "v": 135 },
        { "n": "acidity", "u": "pH", "t": 1235, "v": 7 }
    ]
}
```

Example

Você pode converter valores em uma nova matriz com a seguinte regra.

```
SELECT (SELECT VALUE n FROM e) as sensors FROM 'my/topic'
```

A regra gera a seguinte saída:

```
{
    "sensors": [
        "temperature",
        "light",
        "acidity"
    ]
}
```

Example

Usando a mesma mensagem MQTT, você também pode consultar um valor específico dentro de um objeto aninhado com a seguinte regra.

```
SELECT (SELECT v FROM e WHERE n = 'temperature') as temperature FROM 'my/topic'
```

A regra gera a seguinte saída:

```
{
    "temperature": [
        {
            "v": 22.5
        }
    ]
```

```
}
```

Example

Você também pode nivelar a saída com uma regra mais complicada.

```
SELECT get((SELECT v FROM e WHERE n = 'temperature'), 0).v as temperature FROM 'topic'
```

A regra gera a seguinte saída:

```
{
    "temperature": 22.5
}
```

Como trabalhar com cargas úteis binárias

Quando é necessário tratar a carga útil da mensagem como dados binários brutos, em vez de um objeto JSON, use o operador * para fazer referência a ela em uma cláusula `SELECT`. Isso funciona para cargas úteis que não são JSON com algumas ações de regra, como a ação do S3.

Exemplos de cargas úteis binárias

Quando você usa * para se referir à carga útil da mensagem como dados binários brutos, é possível adicionar dados à regra. Se você tiver uma carga útil vazia ou JSON, a carga útil resultante poderá ter dados adicionados usando a regra. Veja a seguir alguns exemplos de suporte`SELECT`Cláusulas.

- É possível usar o seguinte`SELECT`cláusulas com apenas um * para cargas úteis binárias.
  - ```
    SELECT * FROM 'topic/subtopic'
    ```
  - ```
    SELECT * FROM 'topic/subtopic' WHERE timestamp() % 12 = 0
    ```
- Também é possível adicionar dados e usar o seguinte`SELECT`Cláusulas.
  - ```
    SELECT *, principal() as principal, timestamp() as time FROM 'topic/subtopic'
    ```
  - ```
    SELECT encode(*, 'base64') AS data, timestamp() AS ts FROM 'topic/subtopic'
    ```
- Você também pode usá-los`SELECT`cláusulas com cargas úteis binárias.
  - O seguinte refere-se a`device_type`na cláusula WHERE.

    ```
    SELECT * FROM 'topic/subtopic' WHERE device_type = 'thermostat'
    ```
  - O seguinte também é compatível.

    ```
    {
        "sql": "SELECT * FROM 'topic/subtopic'"
        "actions": [{
            "republish": {
                "topic":"device/${device_id}"
             }
         }]
    }
    ```

As ações de regra a seguir não suportam cargas binárias, portanto, você deve decodificá-las.

- Algumas ações de regra não oferecem suporte a entrada de carga útil binária, como umAção do Lambda, então você deve decodificar cargas binárias. A ação de regra do Lambda poderá receber dados binários, se estiverem codificados em base64 e em uma carga útil JSON. É possível fazer isso alterando a regra para a seguinte.

```
SELECT encode(*, 'base64') AS data FROM 'my_topic'
```

- A instrução SQL não suporta string como entrada. Para converter uma entrada de string em JSON, é possível executar o seguinte comando.

```
SELECT decode(encode(*, 'base64'), 'base64') AS payload FROM 'topic'
```

Versões do SQL

O mecanismo de regras da AWS IoT usa uma sintaxe do tipo SQL para selecionar os dados de mensagens MQTT. As instruções SQL são interpretadas com base em uma versão do SQL especificada com a propriedade `awsIotSqlVersion` em um documento JSON que descreve a regra. Para obter mais informações sobre a estrutura de documentos de regra JSON, consulte Como criar uma regra (p. 481). A propriedade `awsIotSqlVersion` permite especificar qual versão do mecanismo de regras do SQL do AWS IoT você deseja usar. Quando uma nova versão é implantada, você pode continuar usando uma versão mais antiga ou alterar a regra para usar a nova versão. As regras atuais continuam a usar a versão com a qual foram criadas.

O seguinte exemplo de JSON mostra como especificar a versão do SQL usando a propriedade `awsIotSqlVersion`.

```
{
    "sql": "expression",
    "ruleDisabled": false,
    "awsIotSqlVersion": "2016-03-23",
    "actions": [{
        "republish": {
            "topic": "my-mqtt-topic",
            "roleArn": "arn:aws:iam::123456789012:role/my-iot-role"
        }
    }]
}
```

No momento, o AWS IoT oferece suporte às seguintes versões do SQL:

- `2016-03-23`— a versão do SQL implementada em 23/03/2016 (recomendado).
- `2015-10-08`— A versão original do SQL implementada em 08/10/2015.
- `beta`— a versão beta mais recente do SQL. Essa versão pode causar alterações nas regras.

Novidades da versão 23/03/2016 do mecanismo de regras SQL

- Correções na seleção de objetos JSON aninhados.
- Correções em consultas de matrizes.
- Suporte para consulta entre objetos. Para obter mais informações, consulteConsultas de objeto aninhado (p. 629)
- Suporte para emitir uma matriz como um objeto de nível superior.
- Adição da função `encode(value, encodingScheme)`, que pode ser aplicada em dados em formato JSON e não JSON. Para obter mais informações, consulte a função de codificação (p. 594).

Resultado de uma `Array` como um objeto de nível superior

Esse recurso permite que uma regra gere uma matriz como um objeto de nível superior. Por exemplo, com base na seguinte mensagem MQTT:

```
{
    "a": {"b":"c"},
    "arr":[1,2,3,4]
}
```

E a seguinte regra:

```
SELECT VALUE arr FROM 'topic'
```

A regra gera a seguinte saída:

```
[1,2,3,4]
```

Serviço Device Shadow do AWS IoT

O serviço Device Shadow do AWS IoT adiciona sombras a objetos de coisas do AWS IoT. As sombras podem disponibilizar o estado de um dispositivo para aplicativos e outros serviços, independentemente de o dispositivo estar conectado ao AWS IoT ou não. O AWS IoT pode ter várias sombras nomeadas para que sua solução de IoT tenha mais opções para conectar seus dispositivos a outros aplicativos e serviços.

AWS IoTOs objetos de coisas não têm nenhuma sombra nomeada até que sejam criadas explicitamente; no entanto, uma sombra clássica sem nome é criada para uma coisa quando a coisa é criada. As sombras podem ser criadas, atualizadas e excluídas usando o console do AWS IoT. Dispositivos, outros clientes da web e serviços podem criar, atualizar e excluir sombras usando o MQTT e aTópicos MQTT reservados (p. 114), HTTP usando oAPI REST do Device Shadow (p. 659), e oAWS CLIpeloAWS IoT. Porque as sombras são armazenadas porAWSNa nuvem, eles podem coletar e relatar dados de estado do dispositivo de aplicativos e outros serviços na nuvem, independentemente de o dispositivo estar conectado ou não.

Usar shadows

As sombras fornecem um armazenamento de dados confiável para dispositivos, aplicativos e outros serviços na nuvem para compartilhamento de dados. Eles permitem que dispositivos, aplicativos e outros serviços na nuvem se conectem e desconectem sem perder o estado de um dispositivo.

Enquanto dispositivos, aplicativos e outros serviços na nuvem estão conectados ao AWS IoT, eles podem acessar e controlar o estado atual de um dispositivo por meio de suas sombras. Por exemplo, um aplicativo pode solicitar uma alteração no estado de um dispositivo atualizando uma sombra. O AWS IoT publica uma mensagem que indica a alteração no dispositivo. O dispositivo recebe essa mensagem, atualiza seu estado de forma correspondente e publica uma mensagem com o estado atualizado. O serviço Device Shadow reflete esse estado atualizado na sombra correspondente. O aplicativo pode se inscrever na atualização da sombra ou consultar a sombra para saber seu estado atual.

Quando um dispositivo fica offline, um aplicativo ainda pode se comunicar com o AWS IoT e com as sombras do dispositivo. Quando o dispositivo se reconecta, ele recebe o estado atual de suas sombras para que possa atualizar seu estado de forma correspondente a elas, e possa publicar uma mensagem com seu estado atualizado. Da mesma forma, quando o estado do dispositivo muda enquanto ele está offline, o dispositivo mantém a sombra atualizada para que o aplicativo possa consultar as sombras para saber seus estados atuais quando ele se reconectar.

Se seus dispositivos estiverem frequentemente offline e você quiser configurar seus dispositivos para receber mensagens delta depois que eles se reconectarem, você poderá usar o recurso de sessão persistente. Para obter mais informações sobre o período de expiração da sessão persistente, consultePeríodo de expiração da sessão persistente.

Optar por usar sombras nomeadas ou sem nome

O serviço Device Shadow é compatível com sombras clássicas nomeadas e sem nome, como já foram usadas no passado. Um objeto de coisa pode ter várias sombras nomeadas, e não mais do que uma sombra clássica sem nome. Um objeto de coisa pode ter sombras nomeadas e sem nome ao mesmo tempo. No entanto, a API usada para acessar cada uma é ligeiramente diferente, portanto, talvez seja mais eficiente decidir o tipo de sombra que funcionará melhor para sua solução e usar somente esse tipo. Para obter mais informações sobre a API para acessar as sombras, consulte Tópicos de sombra (p. 114).

Com sombras nomeadas, você pode criar diferentes visualizações do estado de um objeto de coisa. Por exemplo, é possível dividir um objeto de coisa com muitas propriedades em sombras com grupos lógicos de propriedades, cada um identificado por seu nome de sombra. Também é possível limitar o acesso às propriedades agrupando-as em diferentes sombras e usando políticas para controlar o acesso. Para obter mais informações sobre políticas a serem usadas com sombras de dispositivos, consulteAções, recursos e chaves de condição para oAWS IoT.

As sombras clássicas e sem nome são mais simples, mas um pouco mais limitadas do que as sombras nomeadas. Cada objeto de coisa do AWS IoT pode ter apenas uma sombra sem nome. Se você espera que sua solução de IoT tenha uma necessidade limitada de dados de sombra, talvez essa seja a maneira como você quer começar a usar sombras. No entanto, se você espera adicionar sombras adicionais no futuro, considere usar sombras nomeadas desde o início.

A indexação de frota suporta sombras sem nome e sombras nomeadas de forma diferente. Para obter mais informações, consulteIndexação de frotas (p. 836).

Acessar sombras

Cada sombra tem um tópico MQTT (p. 114) e um URL HTTP (p. 659) reservados que são compatíveis com as ações `get`, `update` e `delete` na sombra.

As sombras usam documentos JSON de sombra (p. 671) para armazenar e recuperar dados. Um documento de sombra contém uma propriedade de estado que descreve estes aspectos do estado do dispositivo:

- `desired`

  Os aplicativos especificam os estados desejados das propriedades do dispositivo atualizando o objeto `desired`.
- `reported`

  Os dispositivos relatam seu estado atual no objeto `reported`.
- `delta`

  O AWS IoT relata as diferenças entre o estado desejado e relatado no objeto `delta`.

Os dados armazenados em uma sombra são determinados pela propriedade de estado do corpo da mensagem da ação de atualização. As ações de atualização subsequentes podem modificar os valores de um objeto de dados existente e também adicionar e excluir chaves e outros elementos no objeto de estado da sombra. Para obter mais informações sobre como acessar sombras, consulte Usar sombras em dispositivos (p. 637) e Usar sombras em aplicativos e serviços (p. 640).

> **Important**
>
> A permissão para fazer solicitações de atualização deve ser limitada a aplicativos e dispositivos confiáveis. Isso impede que a propriedade de estado da sombra seja alterada inesperadamente. Caso contrário, os dispositivos e aplicativos que usam a sombra devem ser projetados para esperar que as chaves na propriedade de estado sejam alteradas.

Usar sombras em dispositivos, aplicativos e outros serviços na nuvem

O uso de sombras em dispositivos, aplicativos e outros serviços na nuvem requer consistência e coordenação entre todos eles. O serviço Device Shadow do AWS IoT armazena o estado da sombra, envia

mensagens quando o estado da sombra é alterado e responde a mensagens que alteram seu estado. Os dispositivos, aplicativos e outros serviços na nuvem em sua solução de IoT devem gerenciar seu estado e mantê-lo consistente com o estado da sombra do dispositivo.

Os dados de estado de sombra são dinâmicos e podem ser alterados pelos dispositivos, aplicativos e outros serviços na nuvem com permissão para acessar a sombra. Por esse motivo, é importante considerar como cada dispositivo, aplicativo e outro serviço de nuvem interagirão com a sombra. Por exemplo:

- Os dispositivos devem gravar somente na propriedade `reported` do estado de sombra ao comunicar dados de estado à sombra.
- Os aplicativos e outros serviços na nuvem devem gravar somente na propriedade `desired` ao comunicar solicitações de alteração de estado ao dispositivo por meio da sombra.

> **Important**
>
> Os dados contidos em um objeto de dados de sombra são independentes dos de outras sombras e de outras propriedades de objeto de coisa, como os atributos de uma coisa e o conteúdo de mensagens MQTT que um dispositivo de objeto de coisa pode publicar. Um dispositivo pode, no entanto, relatar os mesmos dados em diferentes tópicos MQTT e sombras, se necessário. Um dispositivo compatível com várias sombras deve manter a consistência dos dados que ele relata nas diferentes sombras.

Ordem das mensagens

Não há garantia de que as mensagens do serviço AWS IoT serão enviadas para o dispositivo em uma ordem específica. O cenário a seguir mostra o que acontece nesse caso.

Documento de estado inicial:

```
{
  "state": {
    "reported": {
      "color": "blue"
    }
  },
  "version": 9,
  "timestamp": 123456776
}
```

Atualização 1:

```
{
  "state": {
    "desired": {
      "color": "RED"
    }
  },
  "version": 10,
  "timestamp": 123456777
}
```

Atualização 2:

```
{
  "state": {
```

```
    "desired": {
      "color": "GREEN"
    }
  },
  "version": 11,
  "timestamp": 123456778
}
```

Documento de estado final:

```
{
  "state": {
    "reported": {
      "color": "GREEN"
    }
  },
  "version": 12,
  "timestamp": 123456779
}
```

Isso resulta em duas mensagens delta:

```
{
  "state": {
    "color": "RED"
  },
  "version": 11,
  "timestamp": 123456778
}
```

```
{
  "state": {
    "color": "GREEN"
  },
  "version": 12,
  "timestamp": 123456779
}
```

O dispositivo pode receber essas mensagens fora de ordem. Como o estado nessas mensagens é cumulativo, um dispositivo pode descartar com segurança as mensagens que contêm um número de versão anterior ao que ele está rastreando. Se o dispositivo receber o delta para a versão 12 antes da versão 11, pele poderá descartar a mensagem da versão 11 com segurança.

Reduzir mensagens de shadow

Para reduzir o tamanho das mensagens de shadow enviadas para o dispositivo, defina uma regra que selecione apenas os campos de que o dispositivo precisa e, em seguida, publique novamente a mensagem em um tópico MQTT que o dispositivo esteja escutando.

A regra é especificada no JSON e deve ser semelhante ao seguinte:

```
{
  "sql": "SELECT state, version FROM '$aws/things/+/shadow/update/delta'",
  "ruleDisabled": false,
  "actions": [
    {
      "republish": {
        "topic": "${topic(3)}/delta",
```

```
        "roleArn": "arn:aws:iam:123456789012:role/my-iot-role"
      }
    }
  ]
}
```

A declaração de SELECT determina quais campos da mensagem serão publicados novamente no tópico especificado. Um caractere curinga "+" é usado para corresponder a todos os nomes de shadow. A regra especifica que todas as mensagens correspondentes devem ser publicadas novamente no tópico especificado. Nesse caso, a função `"topic()"` é usada para especificar o tópico no qual a publicação deverá ser refeita. O `topic(3)` avalia o nome da coisa no tópico original. Para obter mais informações sobre como criar regras, consulte Regras para AWS IoT (p. 478).

Usar sombras em dispositivos

Esta seção descreve comunicações do dispositivo com sombras usando mensagens MQTT, o método preferido para dispositivos se comunicarem com o serviço Device Shadow do AWS IoT.

As comunicações de sombra emulam um modelo de solicitação/resposta usando o modelo de comunicação de publicação/assinatura do MQTT. Cada ação de sombra consiste em um tópico de solicitação, um tópico de resposta bem-sucedido (`accepted`) e um tópico de resposta de erro (`rejected`).

Para que os aplicativos e serviços possam determinar se um dispositivo está conectado, consulte Detectar se um dispositivo está conectado (p. 641).

> **Important**
>
> Como o MQTT usa um modelo de comunicação de publicação/assinatura, você deve assinar os tópicos de respostaantesvocê publica um tópico de solicitação. Caso contrário, talvez você não receba a resposta à solicitação que publicar.
> Se você usar umAWS IoT Device SDK (p. 1278)para chamar as APIs de serviço Device Shadow, isso é tratado para você.

Os exemplos desta seção usam uma forma abreviada do tópico em que a*`ShadowTopicPrefix`*pode se referir a uma sombra nomeada ou sem nome, conforme descrito nesta tabela.

As sombras podem ser nomeadas ou sem nome (clássica). Os tópicos usados por cada uma diferem apenas no prefixo do tópico. Esta tabela mostra o prefixo do tópico usado em cada tipo de sombra.

Valor de ***`ShadowTopicPrefix`***	Tipo de sombra
`$aws/things/`*`thingName`*`/shadow`	Sombra sem nome (clássica)
`$aws/things/`*`thingName`*`/shadow/name/`*`shadowName`*	Sombra nomeada

> **Important**
>
> Verifique se o uso de sombras por seu aplicativo ou serviço é consistente e compatível com as implementações correspondentes em seus dispositivos. Por exemplo, considere como as sombras são criadas, atualizadas e excluídas. Considere também como as atualizações são tratadas no dispositivo e nos aplicativos ou serviços que acessam o dispositivo por meio de uma

sombra. Seu design deve ser claro sobre como o estado do dispositivo é atualizado e relatado e como seus aplicativos e serviços interagem com o dispositivo e suas sombras.

Para criar um tópico completo, selecione o *ShadowTopicPrefix* do tipo de sombra ao qual você quer fazer referência, substitua *thingName*, e *shadowName*, se aplicável, por seus valores correspondentes e acrescente isso ao stub de tópico, conforme mostrado na seguinte tabela. Lembre-se de que os tópicos diferenciam maiúsculas de minúsculas.

Consulte Tópicos de sombra (p. 114) para obter mais informações sobre os tópicos reservados para sombras.

Inicializar o dispositivo na primeira conexão ao AWS IoT

Depois que um dispositivo se registra com o AWS IoT, ele deve assinar essas mensagens MQTT para as sombras com as quais é compatível.

Tópico	Significado	Ação que um dispositivo deve executar quando este tópico é recebido
ShadowTopicPrefix/delete/accepted	A solicitação `delete` foi aceita, e o AWS IoT excluiu a sombra.	As ações necessárias para acomodar a sombra excluída, como interromper a publicação de atualizações.
ShadowTopicPrefix/delete/rejected	A solicitação `delete` foi rejeitada pelo AWS IoT e a sombra não foi excluída. O corpo da mensagem contém as informações do erro.	Responda à mensagem de erro no corpo da mensagem.
ShadowTopicPrefix/get/accepted	A solicitação `get` foi aceita pelo AWS IoT, e o corpo da mensagem contém o documento de sombra atual.	As ações necessárias para processar o documento de estado no corpo da mensagem.
ShadowTopicPrefix/get/rejected	A solicitação `get` foi rejeitada pelo AWS IoT, e o corpo da mensagem contém as informações do erro.	Responda à mensagem de erro no corpo da mensagem.
ShadowTopicPrefix/update/accepted	A solicitação `update` foi aceita pelo AWS IoT, e o corpo da mensagem contém o documento de sombra atual.	Confirme se os dados atualizados no corpo da mensagem correspondem ao estado do dispositivo.
ShadowTopicPrefix/update/rejected	A solicitação `update` foi rejeitada pelo AWS IoT, e o corpo da mensagem contém as informações do erro.	Responda à mensagem de erro no corpo da mensagem.
ShadowTopicPrefix/update/delta	O documento de sombra foi atualizado por uma solicitação ao AWS IoT, e o corpo da mensagem contém as alterações solicitadas.	Atualize o estado do dispositivo para que corresponda ao estado desejado no corpo da mensagem.

Tópico	Significado	Ação que um dispositivo deve executar quando este tópico é recebido
ShadowTopicPrefix/update/documents	Uma atualização na sombra foi concluída recentemente e o corpo da mensagem contém o documento de sombra atual.	Confirme se o estado atualizado no corpo da mensagem corresponde ao estado do dispositivo.

Depois de assinar as mensagens na tabela anterior para cada sombra, o dispositivo deve testar para ver se as sombras às quais oferece suporte já foram criadas pela publicação de um tópico `/get` para cada sombra. Se uma mensagem `/get/accepted` for recebida, o corpo da mensagem conterá o documento de sombra que o dispositivo pode usar para inicializar seu estado. Se uma mensagem `/get/rejected` for recebida, a sombra deverá ser criada publicando uma mensagem `/update` com o estado do dispositivo atual.

Por exemplo, suponha que você tem uma coisa`My_IoT_Thing`que não tem sombras clássicas ou nomeadas. Se você publicar agora um`/get`solicitação sobre o tópico reservado`$aws/things/My_IoT_Thing/shadow/get`, retorna um erro na`$aws/things/My_IoT_Thing/shadow/get/rejected`tópico porque a coisa não tem sombras. Primeiro, publique esse erro`/update`mensagem usando o`$aws/things/My_IoT_Thing/shadow/update`tópico com o estado atual do dispositivo, como a seguinte carga útil.

```
 "state": {
    "reported": {
      "welcome": "aws-iot",
      "color": "yellow"
    }
  }
}
```

Uma sombra clássica agora é criada para a coisa e a mensagem é publicada no`$aws/things/My_IoT_Thing/shadow/update/accepted`tópico. Se você publicar no tópico`$aws/things/My_IoT_Thing/shadow/get`, retorna uma resposta ao`$aws/things/My_IoT_Thing/shadow/get/accepted`tópico com o estado do dispositivo.

Para sombras nomeadas, você deve primeiro criar a sombra nomeada ou publicar uma atualização com o nome da sombra antes de usar a solicitação get. Por exemplo, para criar uma shadow nomeada`namedShadow1`, primeiro publique as informações de estado do dispositivo no tópico`$aws/things/My_IoT_Thing/shadow/namedShadow1/update`. Para recuperar as informações de estado, use o`/get`solicitação para a sombra nomeada,`$aws/things/My_IoT_Thing/shadow/namedShadow1/get`.

Processar mensagens enquanto o dispositivo está conectado ao AWS IoT

Enquanto um dispositivo está conectado aoAWS IoT, ele pode receber/update/deltaAs mensagens e devem manter o estado do dispositivo correspondente às alterações em suas sombras:

1. Lendo todas as mensagens /update/delta recebidas e sincronizando o estado do dispositivo para que corresponda.
2. Publicando uma mensagem /update com um corpo de mensagem `reported` que tenha o estado atual do dispositivo, sempre que o estado do dispositivo for alterado.

Enquanto um dispositivo está conectado, ele deve publicar essas mensagens quando indicado.

Indicação	Tópico	Carga útil
O estado do dispositivo foi alterado.	*ShadowTopicPrefix*/update	Um documento de sombra com a propriedade reported.
O dispositivo pode não estar sincronizado com a sombra.	*ShadowTopicPrefix*/get	(empty)
Uma ação no dispositivo indica que uma sombra não será mais compatível com o dispositivo, como quando o dispositivo está sendo removido ou substituído.	*ShadowTopicPrefix*/delete	(empty)

Processar mensagens quando o dispositivo se reconecta ao AWS IoT

Quando um dispositivo com uma ou mais sombras se conecta ao AWS IoT, ele deve sincronizar seu estado com o de todas as sombras com as quais é compatível:

1. Lendo todas as mensagens /update/delta recebidas e sincronizando o estado do dispositivo para que corresponda.
2. Publicando uma mensagem /update com um corpo de mensagem reported que tenha o estado atual do dispositivo.

Usar sombras em aplicativos e serviços

Esta seção descreve como um aplicativo ou serviço interage com o serviço Device Shadow do AWS IoT. Este exemplo pressupõe que o aplicativo ou serviço esteja interagindo apenas com a sombra e, por meio da sombra, com o dispositivo. Este exemplo não inclui nenhuma ação de gerenciamento, como criação ou exclusão de sombras.

Este exemplo usa a API REST do serviço Device Shadow do AWS IoT para interagir com sombras. Ao contrário do exemplo usado no Usar sombras em dispositivos (p. 637), que usa um modelo de comunicação de publicação/assinatura, este exemplo usa o modelo de comunicação de solicitação/resposta da API REST. Isso significa que o aplicativo ou serviço deve fazer uma solicitação para que possa receber uma resposta do AWS IoT. Uma desvantagem deste modelo, no entanto, é que ele não é compatível com notificações. Se seu aplicativo ou serviço exigir notificações oportunas de alterações de estado do dispositivo, considere os protocolos MQTT ou MQTT sobre WSS, que são compatíveis com o modelo de comunicação de publicação/assinatura, conforme descrito em Usar sombras em dispositivos (p. 637).

> **Important**
>
> Verifique se o uso de sombras por seu aplicativo ou serviço é consistente e compatível com as implementações correspondentes em seus dispositivos. Considere, por exemplo, como as sombras são criadas, atualizadas e excluídas e como as atualizações são tratadas no dispositivo e nos aplicativos ou serviços que acessam a sombra. Seu design deve especificar claramente como o estado do dispositivo é atualizado e relatado, e como seus aplicativos e serviços interagem com o dispositivo e suas sombras.

O URL da API REST para sombras nomeadas é:

```
https://endpoint/things/thingName/shadow?name=shadowName
```

e para uma sombra sem nome:

```
https://endpoint/things/thingName/shadow
```

em que:

endpoint

O endpoint retornado pelo comando da CLI:

```
aws iot describe-endpoint --endpoint-type IOT:Data-ATS
```

thingName

O nome do objeto da coisa ao qual a sombra pertence

shadowName

O nome da sombra nomeada. Esse parâmetro não é usado com sombras sem nome.

Inicializar o aplicativo ou serviço na conexão com o AWS IoT

Quando o aplicativo se conecta pela primeira vez ao AWS IoT, ele deve enviar uma solicitação HTTP GET para os URLs das sombras que ele usa para obter o estado atual das sombras que está usando. Isso permite que ele sincronize o aplicativo ou serviço com a sombra.

Processar alterações de estado enquanto o aplicativo ou serviço está conectado ao AWS IoT

Enquanto o aplicativo ou serviço está conectado ao AWS IoT, ele pode consultar o estado atual periodicamente enviando uma solicitação HTTP GET nos URLs das sombras que usa.

Quando um usuário final interage com o aplicativo ou serviço para alterar o estado do dispositivo, o aplicativo ou serviço pode enviar uma solicitação HTTP POST aos URLs das sombras que usa para atualizar o estado `desired` da sombra. Essa solicitação retorna a alteração que foi aceita, mas talvez você precise pesquisar a sombra fazendo solicitações HTTP GET até que o dispositivo atualize a sombra com seu novo estado.

Detectar se um dispositivo está conectado

Para determinar se um dispositivo está conectado no momento, inclua uma propriedade `connected` no documento de sombra e use uma mensagem MQTT Last Will and Testament (LWT) para definir a propriedade `connected` como `false` se um dispositivo for desconectado devido a um erro.

Note

Mensagens MQTT LWT enviadas a tópicos AWS IoT reservados (tópicos que começam com $) são ignoradas pelo serviço Device Shadow do AWS IoT. No entanto, eles são processados por clientes inscritos e pelo mecanismo de regras do AWS IoT, portanto, você precisará criar

uma mensagem LWT que é enviada a um tópico não reservado, e uma regra que republica a mensagem MQTT LWT como uma mensagem de atualização de sombra ao tópico de atualização reservado da sombra, *`ShadowTopicPrefix`*`/update`.

Como enviar ao serviço Device Shadow uma mensagem LWT

1. Crie uma regra que republica a mensagem MQTT LWT no tópico reservado. O exemplo a seguir é uma regra que escuta mensagens na`my/things/myLightBulb/update`tópico e republica para`$aws/things/myLightBulb/shadow/update`.

```
{
    "rule": {
    "ruleDisabled": false,
    "sql": "SELECT * FROM 'my/things/myLightBulb/update'",
    "description": "Turn my/things/ into $aws/things/",
    "actions": [
        {
        "republish": {
            "topic": "$$aws/things/myLightBulb/shadow/update",
            "roleArn": "arn:aws:iam:123456789012:role/aws_iot_republish"
            }
        }
     ]
   }
}
```

2. Quando o dispositivo se conecta ao AWS IoT, ele registra uma mensagem LWT em um tópico não reservado para que seja reconhecido pela regra de republicação. Neste exemplo, esse tópico é `my/things/myLightBulb/update` e define a propriedade conectada como `false`.

```
{
    "state": {
        "reported": {
            "connected":"false"
        }
    }
}
```

3. Após a conexão, o dispositivo publica uma mensagem em seu tópico de atualização de sombra, `$aws/things/myLightBulb/shadow/update`, para relatar seu estado atual, o que inclui a configuração de sua propriedade `connected` como `true`.

```
{
     "state": {
        "reported": {
            "connected":"true"
        }
    }
}
```

4. Antes de desconectar-se corretamente, o dispositivo publica uma mensagem em seu tópico de atualização de sombra, `$aws/things/myLightBulb/shadow/update`, para relatar seu estado mais recente, que inclui a configuração de sua propriedade `connected` como `false`.

```
{
    "state": {
        "reported": {
            "connected":"false"
        }
    }
}
```

5. Se o dispositivo se desconectar devido a um erro, o agente de mensagens do AWS IoT publicará a mensagem LWT do dispositivo em nome do dispositivo. A regra de republicação detecta essa mensagem e publica a mensagem de atualização de sombra para atualizar a `connected` propriedade da sombra do dispositivo.

Simulando comunicações de serviço Device Shadow

Este tópico demonstra como o serviço Device Shadow atua como intermediário e permite que dispositivos e aplicativos usem uma sombra para atualizar, armazenar e recuperar o estado de um dispositivo.

Para demonstrar a interação descrita neste tópico e explorá-la ainda mais, você precisará de umConta da AWSe um sistema em que possa executar oAWS CLI. Se não os tiver, você ainda poderá ver a interação nos exemplos de código.

Neste exemplo, o console do AWS IoT representa o dispositivo. A AWS CLI representa o aplicativo ou serviço que acessa o dispositivo por meio da sombra. A interface da AWS CLI é muito semelhante à API que um aplicativo pode usar para se comunicar com o AWS IoT. O dispositivo neste exemplo é uma lâmpada inteligente, e o aplicativo exibe o estado da lâmpada e pode alterá-lo.

Configurar a simulação

Estes procedimentos inicializam a simulação abrindo o Console do AWS IoT, que simula o dispositivo, e a janela da linha de comando que simula o aplicativo.

Como configurar o ambiente de simulação

1. Criar umConta da AWSOu, se você já tiver um a usar para essa simulação, você pode ignorar esta etapa.

   Você precisará de umConta da AWSPara executar os exemplos deste tópico por conta própria. Se você não tiver umConta da AWS, crie um, conforme descrito emConfigurar seuConta da AWS (p. 19).
2. Abra o Console do AWS IoT, e, no menu esquerdo, escolha Testar para abrir o Cliente MQTT.
3. Em outra janela, abra uma janela de terminal em um sistema que tenha a AWS CLI instalada.

Você deve ter duas janelas abertas: uma com o console do AWS IoT na página Testar e outra com um prompt de linha de comando.

Inicializar o dispositivo

Nesta simulação, trabalharemos com um objeto de coisa nomeado mySimulatedThing e com sua sombra nomeada simShadow1.

Criar objeto coisa e sua política de IoT

Como criar um objeto thing, naAWS IoTConsole do:

1. SelecioneGerenciee, depois, escolhaCoisas.
2. Clique no ícone da barra de ferramentasCriarse as coisas estiverem listadas, caso contrário, cliqueRegistre uma única coisa>Para criar um únicoAWS IoTcoisa.
3. Insira o nome`mySimulatedThing`, deixe outras configurações como padrão e, em seguida, clique emPróximo.

4. Use a criação de certificado com um clique para gerar os certificados que autenticarão a conexão do dispositivo comAWS IoT. Clique emAtivarpara ativar o certificado.
5. É possível anexar a políticaMy_IoT_Policyque daria ao dispositivo permissão para publicar e assinar os tópicos reservados do MQTT. Para obter etapas mais detalhadas sobre como criar umAWS IoTcoisa e como criar essa política, consulteCriar um objeto de coisa (p. 41).

Criar sombra nomeada para o objeto coisa

Você pode criar uma sombra nomeada para uma coisa publicando uma solicitação de atualização no tópico$aws/things/mySimulatedThing/shadow/name/simShadow1/updateconforme descrito abaixo.

Como alternativa, para criar uma sombra nomeada:

1. NoAWS IoTConsole do, escolha seu objeto coisa na lista de coisas exibidas e, em seguida, escolhaSombras.
2. SelecioneAdicionar uma sombra, insira o nomesimShadow1e, depois, escolhaCriarPara adicionar a sombra nomeada.

Assine e publique em tópicos do MQTT reservados

No console, assine os tópicos de sombra MQTT reservados. Esses tópicos são as respostas às ações get, update e delete para que seu dispositivo esteja pronto para receber as respostas depois de publicar uma ação.

Como assinar um tópico MQTT no Cliente MQTT

1. NoCliente MQTT, escolhaAssinar um tópico.
2. Insira oget,update, edeletetópicos para se inscrever. Copie um tópico de cada vez da lista a seguir, cole-o naFiltro de tópicose, depois, clique emSubscribe. Você deve ver os tópicos aparecerem emAssinaturas.

   - $aws/things/mySimulatedThing/shadow/name/simShadow1/delete/accepted
   - $aws/things/mySimulatedThing/shadow/name/simShadow1/delete/rejected
   - $aws/things/mySimulatedThing/shadow/name/simShadow1/get/accepted
   - $aws/things/mySimulatedThing/shadow/name/simShadow1/get/rejected
   - $aws/things/mySimulatedThing/shadow/name/simShadow1/update/accepted
   - $aws/things/mySimulatedThing/shadow/name/simShadow1/update/rejected
   - $aws/things/mySimulatedThing/shadow/name/simShadow1/update/delta
   - $aws/things/mySimulatedThing/shadow/name/simShadow1/update/documents

   Neste ponto, seu dispositivo simulado está pronto para receber os tópicos conforme são publicados pelo AWS IoT.

Como publicar em um tópico MQTT naCliente MQTT

Depois que um dispositivo se inicializar e assinar tópicos de resposta, ele deve consultar as sombras compatíveis. Esta simulação é compatível apenas com uma sombra, a sombra que é compatível com o objeto mySimulatedThing, nomeado como simShadow1.

Como obter o estado da sombra atual no Cliente MQTT

1. No Cliente MQTT, escolha Publicar em um tópico.

2. SobPublicar, insira o tópico a seguir e exclua qualquer conteúdo da janela do corpo da mensagem abaixo do local em que você inseriu o tópico a ser acessado. Então, você pode escolherPublicar no tópicoPara publicar a solicitação.`$aws/things/mySimulatedThing/shadow/name/simShadow1/get`.

   Se você ainda não criou a shadow nomeada,`simShadow1`, você recebe uma mensagem na`$aws/things/mySimulatedThing/shadow/name/simShadow1/get/rejected`tópico `ecode`é`404`, como neste exemplo, uma vez que a sombra não foi criada, portanto, vamos criá-la a seguir.

```
{
  "code": 404,
  "message": "No shadow exists with name: 'simShadow1'"
}
```

Como criar uma sombra com o status atual do dispositivo

1. NoCliente MQTT, escolhaPublicar em um tópicoE insira este tópico:

```
$aws/things/mySimulatedThing/shadow/name/simShadow1/update
```

2. Na janela de corpo da mensagem abaixo do local em que você inseriu o tópico, insira este documento de sombra para mostrar que o dispositivo está relatando seu ID e sua cor atual em valores RGB. SelecionePublicarPara publicar a solicitação.

```
{
  "state": {
    "reported": {
      "ID": "SmartLamp21",
      "ColorRGB": [
        128,
        128,
        128
      ]
    }
  },
  "clientToken": "426bfd96-e720-46d3-95cd-014e3ef12bb6"
}
```

Se você receber uma mensagem no tópico:

- `$aws/things/mySimulatedThing/shadow/name/simShadow1/update/accepted`: Isso significa que a sombra foi criada e o corpo da mensagem contém o documento de sombra atual.
- `$aws/things/mySimulatedThing/shadow/name/simShadow1/update/rejected`: Revise o erro no corpo da mensagem.
- `$aws/things/mySimulatedThing/shadow/name/simShadow1/get/accepted`: A sombra já existe e o corpo da mensagem tem o estado da sombra atual, como neste exemplo. Com isso, você pode definir seu dispositivo ou confirmar que ele corresponde ao estado da sombra.

```
{
  "state": {
    "reported": {
      "ID": "SmartLamp21",
      "ColorRGB": [
        128,
        128,
        128
      ]
```

```
    }
  },
  "metadata": {
    "reported": {
      "ID": {
        "timestamp": 1591140517
      },
      "ColorRGB": [
        {
          "timestamp": 1591140517
        },
        {
          "timestamp": 1591140517
        },
        {
          "timestamp": 1591140517
        }
      ]
    }
  },
  "version": 3,
  "timestamp": 1591140517,
  "clientToken": "426bfd96-e720-46d3-95cd-014e3ef12bb6"
}
```

Enviar uma atualização do aplicativo

Esta seção usa a AWS CLI para demonstrar como um aplicativo pode interagir com uma sombra.

Como obter o estado atual da sombra usando a AWS CLI

Na linha de comando, insira este comando.

```
aws iot-data get-thing-shadow --thing-name mySimulatedThing --shadow-name simShadow1 /dev/
stdout
```

Em plataformas Windows, você pode usar`con`em vez de`/dev/stdout`.

```
aws iot-data get-thing-shadow --thing-name mySimulatedThing --shadow-name simShadow1 con
```

Como a sombra existe e foi inicializada pelo dispositivo para refletir seu estado atual, ela deve retornar o seguinte documento de sombra.

```
{
  "state": {
    "reported": {
      "ID": "SmartLamp21",
      "ColorRGB": [
        128,
        128,
        128
      ]
    }
  },
  "metadata": {
    "reported": {
      "ID": {
        "timestamp": 1591140517
      },
      "ColorRGB": [
```

```
        {
          "timestamp": 1591140517
        },
        {
          "timestamp": 1591140517
        },
        {
          "timestamp": 1591140517
        }
      ]
    }
  },
  "version": 3,
  "timestamp": 1591141111
}
```

O aplicativo pode usar essa resposta para inicializar sua representação do estado do dispositivo.

Se o aplicativo atualizar o estado, como quando um usuário final altera a cor da nossa lâmpada inteligente para amarelo, o aplicativo enviará um comando update-thing-shadow. Esse comando corresponde à API REST `UpdateThingShadow`.

Como atualizar uma sombra de um aplicativo

Na linha de comando, insira este comando.

AWS CLI v2.x

```
aws iot-data update-thing-shadow --thing-name mySimulatedThing --shadow-name simShadow1
 \
    --cli-binary-format raw-in-base64-out \
    --payload '{"state":{"desired":{"ColorRGB":[255,255,0]}},"clientToken":"21b21b21-
bfd2-4279-8c65-e2f697ff4fab"}' /dev/stdout
```

AWS CLI v1.x

```
aws iot-data update-thing-shadow --thing-name mySimulatedThing --shadow-name simShadow1
 \
    --payload '{"state":{"desired":{"ColorRGB":[255,255,0]}},"clientToken":"21b21b21-
bfd2-4279-8c65-e2f697ff4fab"}' /dev/stdout
```

Se for bem-sucedido, esse comando deverá retornar o seguinte documento de sombra.

```
{
  "state": {
    "desired": {
      "ColorRGB": [
        255,
        255,
        0
      ]
    }
  },
  "metadata": {
    "desired": {
      "ColorRGB": [
        {
          "timestamp": 1591141596
        },
        {
          "timestamp": 1591141596
```

```
        },
        {
          "timestamp": 1591141596
        }
      ]
    }
  },
  "version": 4,
  "timestamp": 1591141596,
  "clientToken": "21b21b21-bfd2-4279-8c65-e2f697ff4fab"
}
```

Responder à atualização no dispositivo

Ao retornar ao Cliente MQTT no console da AWS, você deve ver as mensagens que o AWS IoT publicou para refletir o comando de atualização emitido na seção anterior.

Como exibir as mensagens de atualização no Cliente MQTT

No Cliente MQTT, escolha $aws/things/mySimulatedThing/shadow/name/simShadow1/update/delta na coluna Assinaturas. Se o nome do tópico estiver truncado, você poderá pausar nele para ver o tópico completo. No log de tópicos deste tópico, você deve ver um `/delta`mensagem semelhante a esta.

```
{
  "version": 4,
  "timestamp": 1591141596,
  "state": {
    "ColorRGB": [
      255,
      255,
      0
    ]
  },
  "metadata": {
    "ColorRGB": [
      {
        "timestamp": 1591141596
      },
      {
        "timestamp": 1591141596
      },
      {
        "timestamp": 1591141596
      }
    ]
  },
  "clientToken": "21b21b21-bfd2-4279-8c65-e2f697ff4fab"
}
```

Seu dispositivo processará o conteúdo desta mensagem para definir o estado do dispositivo para que corresponda ao estado `desired` na mensagem.

Depois que o dispositivo atualiza o estado para corresponder ao estado `desired` na mensagem, ele deve enviar o novo estado relatado de volta ao AWS IoT publicando uma mensagem de atualização. Este procedimento simula isso no Cliente MQTT.

Como atualizar a sombra no dispositivo

1. No Cliente MQTT, escolha Publicar em um tópico.
2. Na janela do corpo da mensagem, no campo de tópico acima da janela do corpo da mensagem, insira o tópico da sombra seguido pelo `/update`ação:`$aws/things/mySimulatedThing/shadow/`

`name/simShadow1/update`No corpo da mensagem, insira este documento de sombra atualizado, que descreve o estado atual do dispositivo. Clique emPublicarPara publicar o estado do dispositivo atualizado.

```
{
  "state": {
    "reported": {
      "ColorRGB": [255,255,0]
      }
  },
  "clientToken": "a4dc2227-9213-4c6a-a6a5-053304f60258"
}
```

Se a mensagem for recebida com êxito pelo AWS IoT, você deverá ver uma nova resposta no log de mensagens $aws/things/mySimulatedThing/shadow/name/simShadow1/update/accepted no Cliente MQTT com o estado atual da sombra, como neste exemplo.

```
{
  "state": {
    "reported": {
      "ColorRGB": [
        255,
        255,
        0
      ]
    }
  },
  "metadata": {
    "reported": {
      "ColorRGB": [
        {
          "timestamp": 1591142747
        },
        {
          "timestamp": 1591142747
        },
        {
          "timestamp": 1591142747
        }
      ]
    }
  },
  "version": 5,
  "timestamp": 1591142747,
  "clientToken": "a4dc2227-9213-4c6a-a6a5-053304f60258"
}
```

Uma atualização bem-sucedida para o estado relatado do dispositivo também faz com que o AWS IoT envie uma descrição abrangente do estado da sombra em uma mensagem ao tópico , como nesse corpo de mensagem resultante da atualização da sombra executada pelo dispositivo no procedimento anterior.

```
{
  "previous": {
    "state": {
      "desired": {
        "ColorRGB": [
          255,
          255,
          0
        ]
      },
```

```
      "reported": {
        "ID": "SmartLamp21",
        "ColorRGB": [
          128,
          128,
          128
        ]
      }
    },
    "metadata": {
      "desired": {
        "ColorRGB": [
          {
            "timestamp": 1591141596
          },
          {
            "timestamp": 1591141596
          },
          {
            "timestamp": 1591141596
          }
        ]
      },
      "reported": {
        "ID": {
          "timestamp": 1591140517
        },
        "ColorRGB": [
          {
            "timestamp": 1591140517
          },
          {
            "timestamp": 1591140517
          },
          {
            "timestamp": 1591140517
          }
        ]
      }
    },
    "version": 4
  },
  "current": {
    "state": {
      "desired": {
        "ColorRGB": [
          255,
          255,
          0
        ]
      },
      "reported": {
        "ID": "SmartLamp21",
        "ColorRGB": [
          255,
          255,
          0
        ]
      }
    },
    "metadata": {
      "desired": {
        "ColorRGB": [
          {
            "timestamp": 1591141596
          },
```

```
          {
            "timestamp": 1591141596
          },
          {
            "timestamp": 1591141596
          }
        ]
      },
      "reported": {
        "ID": {
          "timestamp": 1591140517
        },
        "ColorRGB": [
          {
            "timestamp": 1591142747
          },
          {
            "timestamp": 1591142747
          },
          {
            "timestamp": 1591142747
          }
        ]
      }
    },
    "version": 5
  },
  "timestamp": 1591142747,
  "clientToken": "a4dc2227-9213-4c6a-a6a5-053304f60258"
}
```

Observar a atualização no aplicativo

O aplicativo agora pode consultar o estado atual da sombra conforme relatado pelo dispositivo.

Como obter o estado atual da sombra usando a AWS CLI

1. Na linha de comando, insira este comando.

   ```
   aws iot-data get-thing-shadow --thing-name mySimulatedThing --shadow-name simShadow1 /
   dev/stdout
   ```

   Em plataformas Windows, você pode usar`con`em vez de`/dev/stdout`.

   ```
   aws iot-data get-thing-shadow --thing-name mySimulatedThing --shadow-name simShadow1
    con
   ```

2. Como a sombra acaba de ser atualizada pelo dispositivo para refletir seu estado atual, ele deve retornar o seguinte documento de sombra.

   ```
   {
     "state": {
       "desired": {
         "ColorRGB": [
           255,
           255,
           0
         ]
       },
       "reported": {
         "ID": "SmartLamp21",
   ```

```
      "ColorRGB": [
        255,
        255,
        0
      ]
    }
  },
  "metadata": {
    "desired": {
      "ColorRGB": [
        {
          "timestamp": 1591141596
        },
        {
          "timestamp": 1591141596
        },
        {
          "timestamp": 1591141596
        }
      ]
    },
    "reported": {
      "ID": {
        "timestamp": 1591140517
      },
      "ColorRGB": [
        {
          "timestamp": 1591142747
        },
        {
          "timestamp": 1591142747
        },
        {
          "timestamp": 1591142747
        }
      ]
    }
  },
  "version": 5,
  "timestamp": 1591143269
}
```

Além da simulação

Experimente com a interação entre oAWS CLI(representando o aplicativo) e o console (representando o dispositivo) para modelar sua solução de IoT.

Interagir com sombras

Este tópico descreve as mensagens associadas a cada um dos três métodos que o AWS IoT fornece para trabalhar com sombras. Esses métodos incluem o seguinte:

`UPDATE`

Cria uma sombra, se ela não existir, ou atualiza o conteúdo de uma sombra existente com as informações de estado fornecidas no corpo da mensagem. O AWS IoT registra um timestamp´com cada atualização para indicar quando o estado foi atualizado pela última vez. Quando o estado da sombra é alterado, o AWS IoT envia mensagens `/delta` a todos os assinantes MQTT com a diferença entre os estados `desired` e `reported`. Os dispositivos ou aplicativos que recebem uma

mensagem `/delta` podem executar ações com base na diferença. Por exemplo, um dispositivo pode atualizar seu estado para o estado desejado, ou um aplicativo pode atualizar sua interface do usuário (UI) para mostrar a alteração no estado do dispositivo.

`GET`

Recupera um documento de sombra atual que contém o estado completo da sombra, incluindo os metadados.

`DELETE`

Exclui a sombra do dispositivo e seu conteúdo.

Não é possível restaurar um documento de sombra de dispositivo excluído, mas é possível criar uma nova shadow de dispositivo usando o nome de um documento de sombra de dispositivo excluído. Se você criar um documento de sombra de dispositivo com o mesmo nome que um que foi excluído nas últimas 48 horas, o número da versão do novo documento de sombra do dispositivo seguirá o do excluído. Se um documento sombra de dispositivo tiver sido excluído por mais de 48 horas, o número da versão de um novo documento sombra de dispositivo com o mesmo nome será 0.

Suporte ao protocolo

O AWS IoT é compatível com o MQTT e uma API REST por meio de protocolos HTTPS para interagir com sombras. O AWS IoT fornece um conjunto de tópicos de solicitação e resposta reservados para ações de publicação e assinatura MQTT. Os dispositivos e os aplicativos devem assinar os tópicos de resposta antes de publicar um tópico de solicitação para obter informações sobre como o AWS IoT tratou a solicitação. Para obter mais informações, consulte Tópicos MQTT do Device Shadow (p. 663) e API REST do Device Shadow (p. 659).

Solicitar e declarar o estado

Ao projetar sua solução de IoT usando o AWS IoT e sombras, você deve determinar os aplicativos ou os dispositivos que solicitarão alterações e aqueles que as implementarão. Normalmente, um dispositivo implementa e relata alterações na sombra e os aplicativos e os serviços respondem e solicitam alterações na sombra. Sua solução pode ser diferente, mas os exemplos neste tópico pressupõem que o aplicativo cliente ou o serviço solicita alterações na sombra e o dispositivo executa as alterações e relata-as de volta à sombra.

Atualizar uma shadow

Seu aplicativo ou serviço pode atualizar o estado de uma sombra usando a API UpdateThingShadow (p. 661) ou publicando no tópico /update (p. 666). As atualizações afetam apenas os campos especificados na solicitação.

Atualizar uma sombra quando um cliente solicita uma alteração de estado

Quando um cliente solicita uma alteração de estado em uma sombra usando o protocolo MQTT

1. O cliente deve ter um documento da sombra atual para que ele possa identificar as propriedades a serem alteradas. Consulte a ação /get para saber como obter o documento de sombra atual.
2. O cliente assina estes tópicos MQTT:

   - `$aws/things/`*`thingName`*`/shadow/name/`*`shadowName`*`/update/accepted`
   - `$aws/things/`*`thingName`*`/shadow/name/`*`shadowName`*`/update/rejected`
   - `$aws/things/`*`thingName`*`/shadow/name/`*`shadowName`*`/update/delta`

- `$aws/things/thingName/shadow/name/shadowName/update/documents`

3. O cliente publica um tópico de solicitação `$aws/things/thingName/shadow/name/shadowName/update` com um documento de estado que contém o estado desejado da sombra. Somente as propriedades a serem alteradas precisam ser incluídas no documento. Este é um exemplo de um documento com o estado desejado.

```
{
  "state": {
    "desired": {
      "color": {
        "r": 10
      },
      "engine": "ON"
    }
  }
}
```

4. Se a solicitação de atualização for válida, o AWS IoT atualizará o estado desejado na sombra e publicará mensagens nestes tópicos:

   - `$aws/things/thingName/shadow/name/shadowName/update/accepted`
   - `$aws/things/thingName/shadow/name/shadowName/update/delta`

   A mensagem `/update/accepted` contém um documento de sombra Documento de estado da resposta /accepted (p. 672), e a mensagem `/update/delta` contém um documento de sombra Documento de estado da resposta /delta (p. 673).

5. Se a solicitação de atualização não for válida, o AWS IoT publicará uma mensagem no tópico `$aws/things/thingName/shadow/name/shadowName/update/rejected` com um documento de sombra Documento de resposta de erro (p. 675) que descreve o erro.

Quando um cliente solicita uma alteração de estado em uma sombra usando a API

1. O cliente chama a API `UpdateThingShadow` (p. 661) com um documento de estado Documento de estado de solicitação (p. 672) como o corpo da mensagem.
2. Se a solicitação for válida, o AWS IoT retornará um código de resposta HTTP bem-sucedido e um documento de sombra Documento de estado da resposta /accepted (p. 672) como o corpo da mensagem de resposta.

   O AWS IoT também publicará uma mensagem MQTT no tópico `$aws/things/thingName/shadow/name/shadowName/update/delta` com um documento de sombra Documento de estado da resposta /delta (p. 673) para quaisquer dispositivos ou clientes que assinaram o tópico.
3. Se a solicitação não foi válida, o AWS IoT retornará um código de resposta de erro HTTP Documento de resposta de erro (p. 675) como o corpo da mensagem de resposta.

Quando o dispositivo recebe o estado `/desired` no tópico `/update/delta`, ele faz as alterações desejadas no dispositivo. Depois, ele envia uma mensagem ao tópico `/update` para relatar seu estado atual para a sombra.

Atualizar uma sombra quando um dispositivo relata seu estado atual

Quando um dispositivo relata seu estado atual para a sombra usando o protocolo MQTT

1. O dispositivo deve assinar estes tópicos MQTT antes de atualizar a sombra:

- `$aws/things/thingName/shadow/name/shadowName/update/accepted`
- `$aws/things/thingName/shadow/name/shadowName/update/rejected`
- `$aws/things/thingName/shadow/name/shadowName/update/delta`
- `$aws/things/thingName/shadow/name/shadowName/update/documents`

2. O dispositivo relata seu estado atual publicando uma mensagem no tópico `$aws/things/thingName/shadow/name/shadowName/update` que relata o estado atual, como neste exemplo.

```
{
    "state": {
        "reported" : {
            "color" : { "r" : 10 },
            "engine" : "ON"
        }
    }
}
```

3. Se o AWS IoT aceitar a atualização, ela publicará uma mensagem nos tópicos `$aws/things/thingName/shadow/name/shadowName/update/accepted` com um documento de sombraDocumento de estado da resposta /accepted (p. 672).
4. Se a solicitação de atualização não for válida, o AWS IoT publicará uma mensagem no tópico `$aws/things/thingName/shadow/name/shadowName/update/rejected` com um documento de sombra Documento de resposta de erro (p. 675) que descreve o erro.

Quando um dispositivo relata seu estado atual para a sombra usando a API

1. O dispositivo chama a API `UpdateThingShadow` (p. 661) com um documento de estado Documento de estado de solicitação (p. 672) como o corpo da mensagem.
2. Se a solicitação for válida, o AWS IoT atualizará a sombra e retornará um código de resposta HTTP de êxito com um documento de sombra Documento de estado da resposta /accepted (p. 672) como o corpo da mensagem de resposta.

   O AWS IoT também publicará uma mensagem MQTT no tópico `$aws/things/thingName/shadow/name/shadowName/update/delta` com um documento de sombra Documento de estado da resposta /delta (p. 673) para quaisquer dispositivos ou clientes que assinaram o tópico.
3. Se a solicitação não foi válida, o AWS IoT retornará um código de resposta de erro HTTP Documento de resposta de erro (p. 675) como o corpo da mensagem de resposta.

Bloqueio otimista

Você pode usar a versão do documento de estado para garantir que está atualizando a versão mais recente de um documento de shadow de dispositivo. Ao fornecer uma versão com uma solicitação de atualização, o serviço rejeitará a solicitação com um código de resposta de conflito HTTP 409 se a versão atual do documento de estado não corresponder à versão fornecida.

Por exemplo:

Documento inicial:

```
{
  "state": {
    "desired": {
      "colors": [
        "RED",
        "GREEN",
```

```
        "BLUE"
      ]
    }
  },
  "version": 10
}
```

Atualização: (versão não corresponde, a solicitação será rejeitada)

```
{
  "state": {
    "desired": {
      "colors": [
        "BLUE"
      ]
    }
  },
  "version": 9
}
```

Resultado:

```
{
  "code": 409,
  "message": "Version conflict",
  "clientToken": "426bfd96-e720-46d3-95cd-014e3ef12bb6"
}
```

Atualização: (a versão corresponde; a solicitação será aceita)

```
{
  "state": {
    "desired": {
      "colors": [
        "BLUE"
      ]
    }
  },
  "version": 10
}
```

Estado final:

```
{
  "state": {
    "desired": {
      "colors": [
        "BLUE"
      ]
    }
  },
  "version": 11
}
```

Recuperar um documento de shadow

Você pode recuperar um documento de sombra usando a API GetThingShadow (p. 660) ou assinando e publicando no tópico /get (p. 664). Isso recupera um documento de sombra completo, incluindo qualquer

delta entre os estados `reported` e `desired`. O procedimento para esta tarefa é o mesmo para um dispositivo ou para um cliente que faz a solicitação.

Como recuperar um documento de sombra usando o protocolo MQTT

1. O dispositivo ou o cliente deve assinar estes tópicos MQTT antes de atualizar a sombra:
   - `$aws/things/thingName/shadow/name/shadowName/get/accepted`
   - `$aws/things/thingName/shadow/name/shadowName/get/rejected`
2. O dispositivo ou o cliente publica uma mensagem no tópico `$aws/things/thingName/shadow/name/shadowName/get` com um corpo de mensagem vazio.
3. Se a solicitação for bem-sucedida, o AWS IoT publica uma mensagem no tópico `$aws/things/thingName/shadow/name/shadowName/get/accepted` com um Documento de estado da resposta /accepted (p. 672) no corpo da mensagem.
4. Se a solicitação não for válida, o AWS IoT publicará uma mensagem no tópico `$aws/things/thingName/shadow/name/shadowName/get/rejected` com um Documento de resposta de erro (p. 675) no corpo da mensagem.

Como recuperar um documento de sombra usando uma API REST

1. O dispositivo ou o cliente chama a API `GetThingShadow (p. 660)` com um corpo de mensagem vazio.
2. Se a solicitação for válida, o AWS IoT retornará um código de resposta HTTP de êxito com um documento de sombra Documento de estado da resposta /accepted (p. 672) como o corpo da mensagem de resposta.
3. Se a solicitação não for válida, o AWS IoT retornará um código de resposta de erro HTTP Documento de resposta de erro (p. 675) como o corpo da mensagem de resposta.

Excluir dados de sombra

Existem duas maneiras de excluir dados de sombra: você pode excluir propriedades específicas no documento de sombra e excluir a sombra completamente.

- Para excluir propriedades específicas de uma sombra, atualize a sombra. No entanto, defina o valor das propriedades que você deseja excluir como `null`. Os campos com um valor de `null` são removidos do documento de sombra.
- Para excluir a sombra inteira, use a API DeleteThingShadow (p. 661) ou publique no tópico /delete (p. 669).

Observe que a exclusão de uma sombra não redefine o número da versão para 0.

Excluir uma propriedade de um documento de sombra

Como excluir uma propriedade de uma sombra usando o protocolo MQTT

1. O dispositivo ou o cliente deve ter um documento de sombra atual para que possa identificar as propriedades a serem alteradas. Consulte Recuperar um documento de shadow (p. 656) para obter informações sobre como obter o documento de sombra atual.
2. O dispositivo ou o cliente assina estes tópicos MQTT:
   - `$aws/things/thingName/shadow/name/shadowName/update/accepted`
   - `$aws/things/thingName/shadow/name/shadowName/update/rejected`

3. O dispositivo ou o cliente publica um tópico de solicitação `$aws/things/thingName/shadow/name/shadowName/update` com um documento de estado que atribui valores `null` às propriedades da sombra a ser excluída. Somente as propriedades a serem alteradas precisam ser incluídas no documento. Este é um exemplo de um documento que exclui a propriedade `engine`.

```
{
  "state": {
    "desired": {
      "engine": null
    }
  }
}
```

4. Se a solicitação de atualização for válida, o AWS IoT excluirá as propriedades especificadas na sombra e publicará uma mensagem no tópico `$aws/things/thingName/shadow/name/shadowName/update/accepted` com um documento de sombra Documento de estado da resposta /accepted (p. 672) no corpo da mensagem.
5. Se a solicitação de atualização não for válida, o AWS IoT publicará uma mensagem no tópico `$aws/things/thingName/shadow/name/shadowName/update/rejected` com um documento de sombra Documento de resposta de erro (p. 675) que descreve o erro.

Como excluir uma propriedade de uma sombra usando a API REST

1. O dispositivo ou o cliente chama a API `UpdateThingShadow (p. 661)` com um Documento de estado de solicitação (p. 672) que atribui valores `null` às propriedades da sombra a ser excluída. Inclua apenas as propriedades que você deseja excluir no documento. Este é um exemplo de um documento que exclui a propriedade `engine`.

```
{
  "state": {
    "desired": {
      "engine": null
    }
  }
}
```

2. Se a solicitação for válida, o AWS IoT retornará um código de resposta HTTP bem-sucedido e um documento de sombra Documento de estado da resposta /accepted (p. 672) como o corpo da mensagem de resposta.
3. Se a solicitação não foi válida, o AWS IoT retornará um código de resposta de erro HTTP Documento de resposta de erro (p. 675) como o corpo da mensagem de resposta.

Excluir uma shadow

> **Note**
>
> Definir o estado da sombra do dispositivo como `null` não exclui a sombra. A versão da sombra será incrementada na próxima atualização.
> A exclusão de uma sobra de dispositivo não exclui o objeto. A exclusão de um objeto não exclui a sombra do dispositivo correspondente.
> A exclusão de uma sombra não redefine o número da versão para 0.

Como excluir uma sombra usando o protocolo MQTT

1. O dispositivo ou o cliente assina estes tópicos MQTT:

   - `$aws/things/thingName/shadow/name/shadowName/delete/accepted`

- `$aws/things/thingName/shadow/name/shadowName/delete/rejected`

2. O dispositivo ou o cliente publica um `$aws/things/thingName/shadow/name/shadowName/delete` com um buffer de mensagens vazio.
3. Se a solicitação de exclusão for válida, o AWS IoT excluirá a sombra e publicará uma mensagem no tópico `$aws/things/thingName/shadow/name/shadowName/delete/accepted` e um documento de sombra Documento de estado da resposta /accepted (p. 672) abreviado no corpo da mensagem. Este é um exemplo de mensagem de exclusão aceita:

```
{
  "version": 4,
  "timestamp": 1591057529
}
```

4. Se a solicitação de atualização não for válida, o AWS IoT publicará uma mensagem no tópico `$aws/things/thingName/shadow/name/shadowName/delete/rejected` com um documento de sombra Documento de resposta de erro (p. 675) que descreve o erro.

Como excluir uma sombra usando a API REST

1. O dispositivo ou o cliente chama a API `DeleteThingShadow (p. 661)` com um buffer de mensagens vazio.
2. Se a solicitação for válida, o AWS IoT retornará um código de resposta HTTP de êxito e um Documento de estado da resposta /accepted (p. 672) e um documento de sombra Documento de estado da resposta /accepted (p. 672) abreviado no corpo da mensagem. Este é um exemplo de mensagem de exclusão aceita:

```
{
  "version": 4,
  "timestamp": 1591057529
}
```

3. Se a solicitação não foi válida, o AWS IoT retornará um código de resposta de erro HTTP Documento de resposta de erro (p. 675) como o corpo da mensagem de resposta.

API REST do Device Shadow

Um shadow expõe a seguinte URI para atualizar as informações de estado:

```
https://account-specific-prefix-ats.iot.region.amazonaws.com/things/thingName/shadow
```

O endpoint é específico para oConta da AWS. Para localizar o endpoint, você pode:

- Usar adescribe-endpointComando daAWS CLI.
- Usar aAWS IoTconfigurações do console. DentroConfigurações, o endpoint está listado emEndpoint personalizado
- Usar aAWS IoTpágina de detalhes da coisa do console. No console do :
  1. AbertoGerenciee emGerencie, escolhaCoisas.
  2. Na lista de coisas, escolha a coisa para a qual você deseja obter o URI do endpoint.
  3. Selecione oSombras de dispositivose escolha sua sombra. Você pode visualizar o URI do endpoint naURL de Device ShadowSeção do doDetalhes Device Shadow.

O formato do endpoint é o seguinte:

```
identifier.iot.region.amazonaws.com
```

A API REST da sombra segue os mesmos mapeamentos de portas/protocolos HTTPS, conforme descrito em Protocolos de comunicação de (p. 83).

> Note
>
> Para usar as APIs, você deve usar`iotdevicegateway`como o nome do serviço para autenticação. Para obter mais informações, consulte`IOTDataPlane`.

Ações da API

- GetThingShadow (p. 660)
- UpdateThingShadow (p. 661)
- DeleteThingShadow (p. 661)
- ListNamedShadowsForThing (p. 662)

Você também pode usar a API para criar uma shadow nomeada fornecendo`name=`*`shadowName`*Como parte do parâmetro de consulta da API.

GetThingShadow

Obtém o shadow da coisa especificada.

O documento de estado de resposta inclui o delta entre os estados `desired` e `reported`.

Solicitação

A solicitação inclui os cabeçalhos HTTP padrão, além da seguinte URI:

```
HTTP GET https://endpoint/things/thingName/shadow?name=shadowName
Request body: (none)
```

O parâmetro de consulta `name` não é necessário para sombras sem nome (clássicas).

Resposta

Após o sucesso, a resposta incluirá os cabeçalhos HTTP padrão e os seguintes código e corpo:

```
HTTP 200
Response Body: response state document
```

Para obter mais informações, consulte Exemplo de documento de estado de resposta (p. 672).

Autorização

Recuperar um shadow requer uma política que permite que o chamador execute a ação `iot:GetThingShadow`. O serviço Device Shadow aceita duas formas de autenticação: Assinatura da versão 4 com credenciais do IAM ou autenticação mútua do TLS com um certificado cliente.

Veja a seguir um exemplo de política que permite que um chamador recupere uma shadow de dispositivo:

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

```
      "Resource": [
        "arn:aws:iot:region:account:thing/thing"
      ]
    }
  ]
}
```

UpdateThingShadow

Atualiza o shadow da coisa especificada.

As atualizações afetam apenas os campos especificados no documento de estado da solicitação. Os campos com um valor de `null` são removido da shadow de dispositivo.

Solicitação

A solicitação inclui os cabeçalhos HTTP padrão e os seguintes URI e corpo:

```
HTTP POST https://endpoint/things/thingName/shadow?name=shadowName
Request body: request state document
```

O parâmetro de consulta `name` não é necessário para sombras sem nome (clássicas).

Para obter mais informações, consulte Exemplo de documento de estado de solicitação (p. 672).

Resposta

Após o sucesso, a resposta incluirá os cabeçalhos HTTP padrão e os seguintes código e corpo:

```
HTTP 200
Response body: response state document
```

Para obter mais informações, consulte Exemplo de documento de estado de resposta (p. 672).

Autorização

Atualizar um shadow requer uma política que permite que o chamador execute a ação `iot:UpdateThingShadow`. O serviço Device Shadow aceita duas formas de autenticação: Assinatura da versão 4 com credenciais do IAM ou autenticação mútua do TLS com um certificado cliente.

Veja a seguir um exemplo de política que permite que um chamador atualize uma shadow de dispositivo:

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

DeleteThingShadow

Apaga o shadow da coisa especificada.

Solicitação

A solicitação inclui os cabeçalhos HTTP padrão, além da seguinte URI:

```
HTTP DELETE https://endpoint/things/thingName/shadow?name=shadowName
Request body: (none)
```

O parâmetro de consulta `name` não é necessário para sombras sem nome (clássicas).

Resposta

Após o sucesso, a resposta incluirá os cabeçalhos HTTP padrão e os seguintes código e corpo:

```
HTTP 200
Response body: Empty response state document
```

Observe que a exclusão de uma sombra não redefine o número da versão para 0.

Autorização

Apagar uma shadow de dispositivo requer uma política que permite que o chamador execute a ação `iot:DeleteThingShadow`. O serviço Device Shadow aceita duas formas de autenticação: Assinatura da versão 4 com credenciais do IAM ou autenticação mútua do TLS com um certificado cliente.

Veja a seguir um exemplo de política que permite que um chamador exclua uma shadow de dispositivo:

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

ListNamedShadowsForThing

Lista as sombras da coisa especificada.

Solicitação

A solicitação inclui os cabeçalhos HTTP padrão, além da seguinte URI:

```
HTTP GET /api/things/shadow/ListNamedShadowsForThing/thingName?
nextToken=nextToken&pageSize=pageSize
Request body: (none)
```

nextToken

O token para recuperação do próximo conjunto de resultados.

Esse valor é retornado nos resultados paginados e é usado na chamada que retorna a próxima página.

pageSize

O número de nomes de sombra a serem retornados em cada chamada. Consulte também `nextToken`.

thingName

O nome da coisa para a qual listar as sombras nomeadas.

Resposta

Quando bem-sucedido, a resposta incluirá os cabeçalhos HTTP padrão e o seguinte código de resposta e umDocumento de resposta da lista de nomes de sombra (p. 675).

Note

A sombra sem nome (clássica) não aparece nesta lista. A resposta é uma lista vazia se você tiver apenas uma sombra clássica ou se `othingName`Você especificar que não existe.

```
HTTP 200
Response body: Shadow name list document
```

Autorização

Listar uma shadow de dispositivo requer uma política que permite que o chamador execute `aiot:ListNamedShadowsForThing`Ação . O serviço Device Shadow aceita duas formas de autenticação: Assinatura da versão 4 com credenciais do IAM ou autenticação mútua do TLS com um certificado cliente.

Veja a seguir um exemplo de política que permite que um chamador liste as sombras nomeadas de uma coisa:

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

Tópicos MQTT do Device Shadow

O serviço Device Shadow usa tópicos MQTT reservados para permitir que aplicativos e dispositivos obtenham, atualizem ou apaguem as informações de estado de um dispositivo (sombra).

Para publicar e inscrever-se em tópicos de shadow, é necessário ter autorização baseada em tópicos. A AWS IoT reserva-se o direito de adicionar novos tópicos à estrutura de tópicos existente. Por esse motivo, recomendamos que você evite assinaturas curinga para os tópicos de shadow. Por exemplo, evite assinar filtros de tópicos como `$aws/things/thingName/shadow/#`, pois o número de tópicos que correspondem a esse filtro de tópicos pode aumentar conforme a AWS IoT inclui novos tópicos de shadow. Para ver exemplos de mensagens publicadas nesses tópicos, consulte Interagir com sombras (p. 652).

As sombras podem ser nomeadas ou sem nome (clássica). Os tópicos usados por cada uma diferem apenas no prefixo do tópico. Esta tabela mostra o prefixo do tópico usado em cada tipo de sombra.

Valor de *ShadowTopicPrefix*	Tipo de sombra
`$aws/things/thingName/shadow`	Sombra sem nome (clássica)
`$aws/things/thingName/shadow/name/shadowName`	Sombra nomeada

Para criar um tópico completo, selecione o *ShadowTopicPrefix* do tipo de sombra ao qual você quer fazer referência, substitua *thingName* e *shadowName*, se aplicável, por seus valores correspondentes e acrescente isso com o stub de tópico, conforme mostrado nas seguintes seções.

Veja a seguir os tópicos do MQTT usados para interagir com shadows.

Tópicos

- /get (p. 664)
- /get/accepted (p. 665)
- /get/rejected (p. 665)
- /update (p. 666)
- /update/delta (p. 667)
- /update/accepted (p. 668)
- /update/documents (p. 668)
- /update/rejected (p. 669)
- /delete (p. 669)
- /delete/accepted (p. 670)
- /delete/rejected (p. 671)

/get

Publique uma mensagem vazia nesse tópico para obter a shadow de dispositivo:

```
ShadowTopicPrefix/get
```

A AWS IoT responde publicando em /get/accepted (p. 665) ou em /get/rejected (p. 665).

Política de exemplo

Veja a seguir um exemplo da política necessária:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iot:Publish"
      ],
      "Resource": [
        "arn:aws:iot:region:account:topic/$aws/things/thingName/shadow/get"
      ]
    }
  ]
}
```

/get/accepted

O AWS IoT publica um documento de sombra de resposta neste tópico ao retornar a sombra do dispositivo:

```
ShadowTopicPrefix/get/accepted
```

Para obter mais informações, consulteDocumentos de estado da resposta (p. 672)

Política de exemplo

Veja a seguir um exemplo da política necessária:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iot:Subscribe"
      ],
      "Resource": [
        "arn:aws:iot:region:account:topicfilter/$aws/things/thingName/shadow/get/accepted"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "iot:Receive"
      ],
      "Resource": [
        "arn:aws:iot:region:account:topic/$aws/things/thingName/shadow/get/accepted"
      ]
    }
  ]
}
```

/get/rejected

A AWS IoT publica um documento do estado do erro nesse tópico quando não é possível gerar a shadow de dispositivo:

```
ShadowTopicPrefix/get/rejected
```

Para obter mais informações, consulteDocumento de resposta de erro (p. 675)

Política de exemplo

Veja a seguir um exemplo da política necessária:

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

```
      "Resource": [
        "arn:aws:iot:region:account:topicfilter/$aws/things/thingName/shadow/get/rejected"
      ]
    },
    {
      "Action": [
        "iot:Receive"
      ],
      "Resource": [
        "arn:aws:iot:region:account:topic/$aws/things/thingName/shadow/get/rejected"
      ]
    }
  ]
}
```

/update

Publique um documento de estado da solicitação nesse tópico para atualizar a shadow de dispositivo:

```
ShadowTopicPrefix/update
```

O corpo da mensagem contém um documento de estado de solicitação parcial (p. 672).

Ao tentar atualizar o estado de um dispositivo, um cliente envia um documento de estado de solicitação JSON com a propriedade `desired` como este:

```
{
  "state": {
    "desired": {
      "color": "red",
      "power": "on"
    }
  }
}
```

Ao atualizar sua sombra, um dispositivo envia um documento de estado de solicitação JSON com a propriedade `reported`, como este:

```
{
  "state": {
    "reported": {
      "color": "red",
      "power": "on"
    }
  }
}
```

A AWS IoT responde publicando em /update/accepted (p. 668) ou em /update/rejected (p. 669).

Política de exemplo

Veja a seguir um exemplo da política necessária:

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

```
      ],
      "Resource": [
        "arn:aws:iot:region:account:topic/$aws/things/thingName/shadow/update"
      ]
    }
  ]
}
```

/update/delta

O AWS IoT publica um documento do estado da resposta nesse tópico quando aceita uma alteração na sombra do dispositivo, e o documento do estado da solicitação contém valores diferentes para os estados `desired` e `reported`:

```
ShadowTopicPrefix/update/delta
```

O buffer de mensagens contém um Documento de estado da resposta /delta (p. 673).

Detalhes do corpo da mensagem

- Uma mensagem publicada em `update/delta` inclui apenas os atributos desejados que diferem entre as seções `desired` e `reported`. Ela contém todos esses atributos, independentemente de esses atributos estarem contidos na mensagem de atualização atual ou de já estarem armazenados na AWS IoT. Os atributos que não diferem entre as seções `desired` e `reported` não são incluídos.
- Se um atributo estiver na seção `reported`, mas não tiver equivalente na seção `desired`, ele não será incluído.
- Se um atributo estiver na seção `desired`, mas não tiver equivalente na seção `reported`, ele será incluído.
- Se um atributo for excluído da seção `reported`, mas ainda existir na seção `desired`, ele será incluído.

Política de exemplo

Veja a seguir um exemplo da política necessária:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iot:Subscribe"
      ],
      "Resource": [
        "arn:aws:iot:region:account:topicfilter/$aws/things/thingName/shadow/update/delta"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "iot:Receive"
      ],
      "Resource": [
        "arn:aws:iot:region:account:topic/$aws/things/thingName/shadow/update/delta"
      ]
    }
  ]
}
```

/update/accepted

A AWS IoT publica um documento do estado da resposta nesse tópico quando aceita uma alteração em uma shadow de dispositivo:

```
ShadowTopicPrefix/update/accepted
```

O buffer de mensagens contém um Documento de estado da resposta /accepted (p. 672).

Política de exemplo

Veja a seguir um exemplo da política necessária:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iot:Subscribe"
      ],
      "Resource": [
        "arn:aws:iot:region:account:topicfilter/$aws/things/thingName/shadow/update/
accepted"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "iot:Receive"
      ],
      "Resource": [
        "arn:aws:iot:region:account:topic/$aws/things/thingName/shadow/update/accepted"
      ]
    }
  ]
}
```

/update/documents

A AWS IoT publica um documento do estado nesse tópico sempre que a atualização de um shadow é feita com êxito:

```
ShadowTopicPrefix/update/documents
```

O corpo da mensagem contém um Documento de estado da resposta /documents (p. 673).

Política de exemplo

Veja a seguir um exemplo da política necessária:

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

```
      ],
      "Resource": [
        "arn:aws:iot:region:account:topicfilter/$aws/things/thingName/shadow/update/
documents"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "iot:Receive"
      ],
      "Resource": [
        "arn:aws:iot:region:account:topic/$aws/things/thingName/shadow/update/documents"
      ]
    }
  ]
}
```

/update/rejected

A AWS IoT publica um documento do estado do erro nesse tópico quando rejeita uma alteração em uma shadow de dispositivo:

```
ShadowTopicPrefix/update/rejected
```

O corpo da mensagem contém um Documento de resposta de erro (p. 675).

Política de exemplo

Veja a seguir um exemplo da política necessária:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iot:Subscribe"
      ],
      "Resource": [
        "arn:aws:iot:region:account:topicfilter/$aws/things/thingName/shadow/update/
rejected"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "iot:Receive"
      ],
      "Resource": [
        "arn:aws:iot:region:account:topic/$aws/things/thingName/shadow/update/rejected"
      ]
    }
  ]
}
```

/delete

Para excluir uma shadow de dispositivo, publique uma mensagem vazia no tópico de exclusão:

```
ShadowTopicPrefix/delete
```

O conteúdo da mensagem é ignorado.

Observe que a exclusão de uma sombra não redefine o número da versão para 0.

A AWS IoT responde publicando em /delete/accepted (p. 670) ou em /delete/rejected (p. 671).

Política de exemplo

Veja a seguir um exemplo da política necessária:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iot:Publish"
      ],
      "Resource": [
        "arn:aws:iot:region:account:topic/$aws/things/thingName/shadow/delete"
      ]
    }
  ]
}
```

/delete/accepted

A AWS IoT publicará uma mensagem nesse tópico quando uma shadow de dispositivo for excluída:

```
ShadowTopicPrefix/delete/accepted
```

Política de exemplo

Veja a seguir um exemplo da política necessária:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iot:Subscribe"
      ],
      "Resource": [
        "arn:aws:iot:region:account:topicfilter/$aws/things/thingName/shadow/delete/
accepted"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "iot:Receive"
      ],
      "Resource": [
        "arn:aws:iot:region:account:topic/$aws/things/thingName/shadow/delete/accepted"
      ]
    }
```

```
  ]
}
```

/delete/rejected

A AWS IoT publica um documento do estado do erro nesse tópico quando não é possível excluir a shadow de dispositivo:

```
ShadowTopicPrefix/delete/rejected
```

O corpo da mensagem contém um Documento de resposta de erro (p. 675).

Política de exemplo

Veja a seguir um exemplo da política necessária:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iot:Subscribe"
      ],
      "Resource": [
        "arn:aws:iot:region:account:topicfilter/$aws/things/thingName/shadow/delete/
rejected"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "iot:Receive"
      ],
      "Resource": [
        "arn:aws:iot:region:account:topic/$aws/things/thingName/shadow/delete/rejected"
      ]
    }
  ]
}
```

Documentos do serviço Device Shadow

O serviço Device Shadow respeita todas as regras da especificação JSON. Os valores, objetos e matrizes são armazenados no documento de shadow do dispositivo.

Índice

- Exemplos de documentos de sombra (p. 672)
- Propriedades do documento (p. 676)
- Estado delta (p. 677)
- Versionamento de documentos de sombra (p. 678)
- Tokens de cliente em documentos de sombra (p. 678)
- Propriedades vazias do documento de sombra (p. 678)
- Valores de matriz em documentos de sombra (p. 679)

Exemplos de documentos de sombra

O serviço Device Shadow usa estes documentos nas operações UPDATE, GET e DELETE usando a API REST (p. 659) ou as Mensagens de publicação/assinatura do MQTT (p. 663).

Exemplos

- Documento de estado de solicitação (p. 672)
- Documentos de estado da resposta (p. 672)
- Documento de resposta de erro (p. 675)
- Documento de resposta da lista de nomes de sombra (p. 675)

Documento de estado de solicitação

Um documento de estado de solicitação tem o seguinte formato:

```
{
    "state": {
        "desired": {
            "attribute1": integer2,
            "attribute2": "string2",
            ...
            "attributeN": boolean2
        },
        "reported": {
            "attribute1": integer1,
            "attribute2": "string1",
            ...
            "attributeN": boolean1
        }
    },
    "clientToken": "token",
    "version": version
}
```

- `state` — As atualizações afetam apenas os campos especificados. Normalmente, você usará a propriedade `reported` ou `desired`, mas não ambas na mesma solicitação.
  - `desired`— as propriedades de estado e os valores solicitados para serem atualizados no dispositivo.
  - `reported`— as propriedades de estado e os valores relatados pelo dispositivo.
- `clientToken`— Se usado, é possível combinar a solicitação e a resposta correspondente pelo token cliente.
- `version` — Se usado, o serviço Device Shadow processará a atualização somente se a versão especificada corresponder à versão mais recente que ele tem.

Documentos de estado da resposta

Os documentos de estado de resposta têm o seguinte formato, dependendo do tipo de resposta.

Documento de estado da resposta /accepted

```
{
    "state": {
        "desired": {
            "attribute1": integer2,
            "attribute2": "string2",
            ...
```

```
            "attributeN": boolean2
        }
    },
    "metadata": {
        "desired": {
            "attribute1": {
                "timestamp": timestamp
            },
            "attribute2": {
                "timestamp": timestamp
            },
            ...
            "attributeN": {
                "timestamp": timestamp
            }
        }
    },
    "timestamp": timestamp,
    "clientToken": "token",
    "version": version
}
```

Documento de estado da resposta /delta

```
{
    "state": {
        "attribute1": integer2,
        "attribute2": "string2",
        ...
        "attributeN": boolean2
        }
    },
    "metadata": {
        "attribute1": {
            "timestamp": timestamp
        },
        "attribute2": {
            "timestamp": timestamp
        },
        ...
        "attributeN": {
            "timestamp": timestamp
        }
    },
    "timestamp": timestamp,
    "clientToken": "token",
    "version": version
}
```

Documento de estado da resposta /documents

```
{
  "previous" : {
    "state": {
        "desired": {
            "attribute1": integer2,
            "attribute2": "string2",
            ...
            "attributeN": boolean2
        },
        "reported": {
            "attribute1": integer1,
            "attribute2": "string1",
```

```
            ...
            "attributeN": boolean1
        }
    },
    "metadata": {
        "desired": {
            "attribute1": {
                "timestamp": timestamp
            },
            "attribute2": {
                "timestamp": timestamp
            },
            ...
            "attributeN": {
                "timestamp": timestamp
            }
        },
        "reported": {
            "attribute1": {
                "timestamp": timestamp
            },
            "attribute2": {
                "timestamp": timestamp
            },
            ...
            "attributeN": {
                "timestamp": timestamp
            }
        }
    },
    "version": version-1
  },
  "current": {
    "state": {
        "desired": {
            "attribute1": integer2,
            "attribute2": "string2",
            ...
            "attributeN": boolean2
        },
        "reported": {
            "attribute1": integer2,
            "attribute2": "string2",
            ...
            "attributeN": boolean2
        }
    },
    "metadata": {
        "desired": {
            "attribute1": {
                "timestamp": timestamp
            },
            "attribute2": {
                "timestamp": timestamp
            },
            ...
            "attributeN": {
                "timestamp": timestamp
            }
        },
        "reported": {
            "attribute1": {
                "timestamp": timestamp
            },
            "attribute2": {
                "timestamp": timestamp
```

```
            },
            ...
            "attributeN": {
                "timestamp": timestamp
            }
        }
    },
    "version": version
  },
  "timestamp": timestamp,
  "clientToken": "token"
}
```

Propriedades do documento de estado de resposta

- `previous`— após uma atualização bem-sucedida, contém o`state`do objeto antes da atualização.
- `current`— após uma atualização bem-sucedida, contém o`state`do objeto após a atualização.
- `state`
  - `reported`— Presente somente se uma coisa reportar algum dado no`reported`e contém apenas campos que estavam no documento de estado da solicitação.
  - `desired`— Presente somente se um dispositivo reportar algum dado no`desired`e contém apenas campos que estavam no documento de estado da solicitação.
  - `delta`— Presente somente se o`desired`Os dados são diferentes da atual`reported`dados.
- `metadata` — Contém as marcações de data e hora de cada atributo nas seções `desired` e `reported` para que você possa determinar quando o estado foi atualizado.
- `timestamp` — A data e a hora de Epoch em que a resposta foi gerada pelo AWS IoT.
- `clientToken` — Presente somente se um token cliente foi usado ao publicar um JSON válido no tópico `/update`.
- `version` — A versão atual do documento da shadow de dispositivo compartilhado na AWS IoT. Ela é incrementada em um em relação à versão anterior do documento.

Documento de resposta de erro

Um documento de resposta de erro tem o seguinte formato:

```
{
    "code": error-code,
    "message": "error-message",
    "timestamp": timestamp,
    "clientToken": "token"
}
```

- `code` — Um código de resposta HTTP que indica o tipo de erro.
- `message` — Uma mensagem de texto que fornece informações adicionais.
- `timestamp` — A data e a hora em que a resposta foi gerada pelo AWS IoT. Esta propriedade não está presente em todos os documentos de resposta do erro.
- `clientToken`— Presente somente se um token cliente foi usado na mensagem publicada.

Para obter mais informações, consulteMensagens de erro do Device Shadow (p. 680)

Documento de resposta da lista de nomes de sombra

Um documento de resposta de lista de nomes de sombra tem o seguinte formato:

```
{
    "results": [
        "shadowName-1",
        "shadowName-2",
        "shadowName-3",
        "shadowName-n"
    ],
    "nextToken": "nextToken",
    "timestamp": timestamp
}
```

- `results`— A matriz de nomes de sombras.
- `nextToken`— O valor do token a ser usado em solicitações paginadas para obter a próxima página na sequência. Essa propriedade não está presente quando não há mais nomes de sombra para retornar.
- `timestamp` — A data e a hora em que a resposta foi gerada pelo AWS IoT.

Propriedades do documento

Um documento de shadow de dispositivo tem as seguintes propriedades:

`state`

`desired`

O estado desejado do dispositivo. Os aplicativos podem gravar nessa parte do documento para atualizar o estado de um dispositivo diretamente sem a necessidade de conectar-se a ele.

`reported`

O estado relatado da coisa. Os dispositivos gravam nessa parte do documento para relatar seu novo estado. Os aplicativos leem essa parte do documento para determinar o último estado relatado do dispositivo.

`metadata`

Informações sobre os dados armazenados na seção `state` do documento. Isso inclui marcações de data e hora, no tempo Epoch, para cada atributo na seção `state`, que permite determinar quando foram atualizados.

> **Note**
>
> Os metadados não contribuem para o tamanho do documento para Service Limits ou definição de preço. Para obter mais informações, consulte AWS IoTLimites de serviço da.

`timestamp`

Indica quando a mensagem foi enviada pelo AWS IoT. Com o uso do timestamp na mensagem e os timestamps de atributos individuais na seção `desired` ou `reported`, um dispositivo pode determinar a idade de uma propriedade, mesmo que o dispositivo não tenha um relógio interno.

`clientToken`

Uma string exclusiva do dispositivo que permite associar respostas com as solicitações em um ambiente MQTT.

`version`

A versão do documento. Toda vez que o documento é atualizado, o número de versão é incrementado. Usado para garantir que a versão do documento que está sendo atualizado seja a mais recente.

Para obter mais informações, consulteExemplos de documentos de sombra (p. 672)

Estado delta

Estado delta é um tipo virtual de estado que contém a diferença entre os estados `desired` e `reported`. Os campos na seção `desired` que não estão na seção `reported` são incluídos no delta. Os campos na seção `reported` e que não estão na seção `desired` não são incluídos no delta. O delta contém metadados, e seus valores são iguais aos metadados no campo `desired`. Por exemplo:

```
{
  "state": {
    "desired": {
      "color": "RED",
      "state": "STOP"
    },
    "reported": {
      "color": "GREEN",
      "engine": "ON"
    },
    "delta": {
      "color": "RED",
      "state": "STOP"
    }
  },
  "metadata": {
    "desired": {
      "color": {
        "timestamp": 12345
      },
      "state": {
        "timestamp": 12345
      }
      },
      "reported": {
        "color": {
          "timestamp": 12345
        },
        "engine": {
          "timestamp": 12345
        }
      },
      "delta": {
        "color": {
          "timestamp": 12345
        },
        "state": {
          "timestamp": 12345
        }
      }
    },
    "version": 17,
    "timestamp": 123456789
  }
}
```

Quando os objetos aninhados diferem, o delta contém o caminho para a raiz.

```
{
  "state": {
    "desired": {
      "lights": {
        "color": {
          "r": 255,
          "g": 255,
          "b": 255
```

```
        }
      }
    },
    "reported": {
      "lights": {
        "color": {
          "r": 255,
          "g": 0,
          "b": 255
        }
      }
    },
    "delta": {
      "lights": {
        "color": {
          "g": 255
        }
      }
    }
  },
  "version": 18,
  "timestamp": 123456789
}
```

O serviço Device Shadow calcula o delta Percorrendo cada campo no estado `desired` e comparando-o com o estado `reported`.

As matrizes são tratadas como valores. Se uma matriz na seção `desired` não corresponder à matriz na seção `reported`, toda a matriz desejada será copiada para o delta.

Versionamento de documentos de sombra

O serviço Device Shadow é compatível com o versionamento em cada mensagem de atualização, tanto de solicitação quanto de resposta. Isso significa que a cada atualização de uma sombra, a versão do documento JSON é incrementada. Isso garante duas coisas:

- Um cliente poderá receber um erro se tentar substituir um shadow usando um número de versão mais antigo. O cliente é informado de que deve sincronizar novamente antes de atualizar uma shadow de dispositivo.
- Um cliente poderá decidir não atuar em uma mensagem recebida se a mensagem tiver uma versão inferior à versão armazenada pelo cliente.

Um cliente pode ignorar a correspondência de versão não incluindo uma versão no documento de sombra.

Tokens de cliente em documentos de sombra

Você pode usar um token cliente com um sistema de mensagens com base em MQTT para verificar se o mesmo token cliente está contido em uma solicitação e em uma resposta de solicitação. Isso garante que a resposta e a solicitação estejam associadas.

> **Note**
>
> O token do cliente pode ter até 64 bytes. Um token de cliente com mais de 64 bytes resulta em uma resposta 400 (Solicitação inválida) e na mensagem de erro clientToken inválido.

Propriedades vazias do documento de sombra

As propriedades `reported` e `desired` em um documento de sombra podem estar vazias ou omitidas quando não se aplicam ao estado da sombra atual. Por exemplo, um documento de sombra contém uma

propriedade `desired` somente se tiver um estado desejado. Veja a seguir um exemplo válido de um documento de estado sem a propriedade `desired`:

```
{
    "reported" : { "temp": 55 }
}
```

A propriedade `reported` também pode estar vazia, como se a sombra não tivesse sido atualizada pelo dispositivo:

```
{
    "desired" : { "color" : "RED" }
}
```

Se uma atualização fizer com que as propriedades `desired` ou `reported` se tornem nulas, ela será removida do documento. Veja a seguir como remover a propriedade `desired` definindo-a como `null`. Você pode fazer isso quando um dispositivo atualiza seu estado, por exemplo.

```
{
    "state": {
        "reported": {
            "color": "red"
        },
        "desired": null
    }
}
```

Um documento de sombra também pode ter nenhuma propriedade `desired` ou `reported`, tornando o documento de sombra vazio. Este é um exemplo de um documento de sombra vazio, mas válido.

```
{
}
```

Valores de matriz em documentos de sombra

As shadows dão suporte a matrizes, mas as trata como valores normais em que uma atualização em uma matriz substitui toda a matriz. Não é possível atualizar parte de uma matriz.

Estado inicial:

```
{
    "desired" : { "colors" : ["RED", "GREEN", "BLUE" ] }
}
```

Atualização:

```
{
    "desired" : { "colors" : ["RED"] }
}
```

Estado final:

```
{
    "desired" : { "colors" : ["RED"] }
}
```

As matrizes não podem ter valores nulos. Por exemplo, a matriz a seguir não é válida e será rejeitada.

```
{
    "desired" : {
        "colors" : [ null, "RED", "GREEN" ]
    }
}
```

Mensagens de erro do Device Shadow

O serviço Device Shadow publica uma mensagem no tópico de erro (por MQTT) quando uma tentativa de alterar o documento de estado falha. Essa mensagem só é emitida como uma resposta a uma solicitação de publicação em um dos reservados`$aws`tópicos. Se o cliente atualizar o documento usando a API REST, ele receberá o código de erro HTTP como parte da resposta, e nenhuma mensagem de erro MQTT será emitida.

Código de erro HTTP	Mensagens de erro
400 (Solicitação inválida)	• JSON inválido • Nó obrigatório ausente: estado • O nó de estado deve ser um objeto • O nó desejado deve ser um objeto • O nó informado deve ser um objeto • Versão inválida • Token cliente inválido Note Um token de cliente com mais de 64 bytes gerará essa resposta. • O JSON contém muitos níveis de aninhamento; o máximo é 6 • O estado contém um nó inválido
401 (Não autorizado)	• Não autorizado
403 (Proibido)	• Proibido
404 (Não encontrado)	• Coisa não encontrada • Não existe nenhuma sombra com o nome: *`shadowName`*
409 (Conflito)	• Conflito de versão
413 (Carga útil muito grande)	• A carga útil excede o tamanho máximo permitido
415 (Tipo de mídia incompatível)	• Codificação documentada incompatível; a codificação compatível é UTF-8
429 (Muitas solicitações)	• O serviço Device Shadow gerará essa mensagem de erro quando houver mais de 10 solicitações em processamento em uma única conexão. Uma solicitação em voo é uma solicitação em andamento que foi iniciada, mas ainda não concluída.
500 (Erro do servidor interno)	• Falha no serviço interno

Trabalhos

Usar oAWS IoTTrabalhos para definir um conjunto de operações remotas que podem ser enviadas e executadas em um ou mais dispositivos conectados àAWS IoT. Por exemplo, você pode definir um trabalho que instrui um conjunto de dispositivos a baixar e instalar aplicativos, executar atualizações de firmware, reinicializar, alternar certificados ou executar operações de solução de problemas remotamente.

Como acessar oAWS IoTempregos

Você pode começar comAWS IoTTrabalhos usando o console do ou oAWS IoT CoreAPI.

Usar o console

Faça login noAWS Management Consolee vá para oAWS IoTconsole do . No painel de navegação, escolha Manage (Gerenciar) e, depois, Jobs (Trabalhos). Você pode criar e gerenciar trabalhos desta seção. Se você quiser criar e gerenciar modelos de trabalho, no painel de navegação, escolhaTemplates de trabalho. Para obter mais informações, consulte Crie e gerencie trabalhos usando oAWS Management Console (p. 690).

Usando a API ou CLI

Você pode começar usando oAWS IoT CoreOperações de API do. Para obter mais informações, consulte Referência da API do AWS IoT. OAWS IoT CoreAPI queAWS IoTjobs é construído em é suportado peloAWSSDK. Para obter mais informações, consulteAWSSDKs e toolkits.

Você pode usar oAWS CLIPara executar comandos para criar e gerenciar trabalhos e modelos de trabalho. Para obter mais informações, consulteAWS IoTReferência da CLI do.

AWS IoTRegiões de trabalhos e endpoints

AWS IoTJobs suportam pontos de extremidade de API de plano de controle e plano de dados específicos para o seuRegião da AWS. Os endpoints da API do plano de dados são específicos para oConta da AWSeRegião da AWS. Para obter mais informações sobre oAWS IoTEndpoints de trabalhos, consulteAWS IoT Device Management- endpoints de dados de trabalhosnoAWSReferência geral.

O que é oAWS IoTTrabalhos do?

Usar oAWS IoTTrabalhos para definir um conjunto de operações remotas que podem ser enviadas e executadas em um ou mais dispositivos conectados àAWS IoT.

Para criar trabalhos, primeiro defina umdocumento de trabalhoque contém uma lista de instruções que descrevem as operações que o dispositivo deve executar remotamente. Para executar essas operações, especifique uma lista detem como alvo, que são coisas individuais,Grupos de coisas (p. 272), ou ambos. O documento de trabalho e as metas em conjunto constituem umimplantação.

Cada implantação pode ter configurações adicionais:

- Distribuição: Essa configuração define quantos dispositivos recebem o documento de trabalho a cada minuto.
- Abort: Se um determinado número de dispositivos não receber a notificação de trabalho, use essa configuração para cancelar o trabalho e evite enviar uma atualização incorreta para uma frota inteira.
- Timeout (Tempo limite): Se uma resposta não for recebida de seus destinos de trabalho dentro de um determinado período, o trabalho poderá falhar. Você pode rastrear o trabalho que está sendo executado nesses dispositivos.
- Tentar novamente: Se um dispositivo relatar falha ou um trabalho expirar, você poderá usarAWS IoTTrabalhos para reenviar o documento de trabalho para o dispositivo automaticamente.

O serviço Jobs da AWS IoT envia uma mensagem para informar aos destinos que um trabalho está disponível. O destino inicia a execução do trabalho fazendo download do documento de trabalho, executando as operações especificadas e relatando o andamento para a AWS IoT. Você pode rastrear o andamento de um trabalho para um alvo específico ou para todos os destinos executando comandos fornecidos porAWS IoTTrabalhos do. Quando um trabalho começa, ele tem um status deEm andamento. Em seguida, os dispositivos relatam atualizações incrementais enquanto exibem esse status até que o trabalho seja bem-sucedido, falhe ou expire o tempo limite.

Os tópicos a seguir descrevem alguns conceitos-chave de trabalhos e o ciclo de vida de trabalhos e execuções de trabalhos.

Tópicos

- Principais conceitos sobre trabalhos (p. 682)
- Estados de execução de trabalhos e trabalhos (p. 684)

Principais conceitos sobre trabalhos

Os conceitos a seguir podem ajudá-lo a entender mais sobreAWS IoTTrabalhos e como criar e implantar trabalhos para executar operações remotas em seus dispositivos.

Conceitos básicos

A seguir, são conceitos básicos que você precisará saber ao usarAWS IoTTrabalhos do.

Trabalho

Um trabalho é uma operação remota que é enviada e executada em um ou mais dispositivos conectados àAWS IoT. Por exemplo, você pode definir um trabalho que instrui um conjunto de dispositivos a baixar e instalar um aplicativo ou executar atualizações de firmware, reinicializar, alternar certificados ou executar operações de solução de problemas remotamente.

documento de Job

Para criar um trabalho, você deve primeiro criar um documento de trabalho, que consiste na descrição das operações remotas a serem realizadas pelos dispositivos.

Os documentos de Job são documentos JSON codificados em UTF-8 e contêm informações das quais seus dispositivos precisam para executar um trabalho. Um documento de trabalho contém um ou mais URLs onde o dispositivo pode fazer download de uma atualização ou de outros dados. O documento de trabalho pode ser armazenado em um bucket do Amazon S3 ou ser incluído em linha com o comando que cria o trabalho.

Tip

Para obter exemplos de documentos de trabalho, consulte o exemplo jobs-agent.js no AWS IoT SDK para JavaScript.

Destino

Ao criar um trabalho, você especifica uma lista de destinos, que são os dispositivos que devem executar as operações. Os destinos podem ser coisas, grupos de coisas (p. 272) ou ambos. O serviço Trabalhos do AWS IoT envia uma mensagem a cada destino para informar que um trabalho está disponível.

Implantação

Depois de criar um trabalho fornecendo o documento de trabalho e especificando sua lista de destinos, o documento de trabalho é implantado nos dispositivos de destino remoto para os quais você deseja executar a atualização. Para trabalhos de snapshot, o trabalho será concluído após a implantação nos dispositivos de destino. Para trabalhos contínuos, um trabalho é implantado em um grupo de dispositivos conforme são adicionados aos grupos.

Trabalhoexecução

Uma execução de trabalho é uma instância de um trabalho em um dispositivo de destino. O destino começa a execução de um trabalho ao fazer download do documento de trabalho. Ele executa as operações especificadas no documento e relata seu progresso ao AWS IoT. Um número de execução é um identificador exclusivo da execução de um trabalho em um destino específico. OAWS IoTO serviço Jobs fornece comandos para acompanhar o andamento de uma execução de trabalho em um destino e o andamento de um trabalho em todos os destinos.

Conceitos de tipos de Job

Os conceitos a seguir podem ajudá-lo a entender mais sobre os diferentes tipos de trabalhos que você pode criar comAWS IoTTrabalhos do.

trabalho de snapshot

Por padrão, um trabalho é enviado para todos os destinos que você especifica ao criar o trabalho. Depois que esses destinos concluem o trabalho (ou relatam que não puderam concluí-lo), o trabalho será concluído.

Continuoustrabalho

Um trabalho contínuo é enviado a todos os destinos que você especifica ao criar o trabalho. Ele continua a executar e é enviado a todos os novos dispositivos (coisas) que são adicionados ao grupo de destino. Por exemplo, um trabalho contínuo pode ser usado para integrar ou atualizar dispositivos conforme são adicionados a um grupo. Você pode criar um trabalho contínuo definindo um parâmetro opcional ao criá-lo.

> **Note**
>
> Ao direcionar sua frota de IoT usando grupos de coisas dinâmicas, recomendamos que você use trabalhos contínuos em vez de trabalhos instantâneos. Ao usar trabalhos contínuos, os dispositivos que se juntam ao grupo recebem a execução do trabalho mesmo após a criação do trabalho.

Pre-signed URLs

Para acesso limitado aos dados seguros que não são incluídos no documento de trabalho, você pode usar pre-signed URLs do Amazon S3. Coloque seus dados em um bucket do Amazon S3 e adicione um link de espaço reservado aos dados no documento de trabalho. QuandoAWS IoTOs trabalhos do recebem uma solicitação para o documento de trabalho, ele analisa o documento de trabalho à procura dos links de espaço reservado e substitui esses links por pre-signed URLs do Amazon S3.

O link de espaço reservado está no formato a seguir:

```
${aws:iot:s3-presigned-url:https://s3.amazonaws.com/bucket/key}
```

em que *bucket* é o nome do bucket e *key* é o objeto no bucket ao qual você está vinculando.

Nas regiões de Pequim e Ningxia, URLs pré-assinados funcionam somente se o proprietário do recurso tem uma licença ICP (Internet Content Provider) (Internet Content Provider). Para obter mais informações, consulteAmazon Simple Storage ServicenoConceitos básicos doAWSServiços na Chinadocumentação.

Conceitos de configuração de Job

Os conceitos a seguir podem ajudá-lo a entender como configurar trabalhos.

Distribuições

Você pode especificar a rapidez com que os destinos são notificados sobre a execução de um trabalho pendente. Isso permite que você crie uma distribuição em etapas para gerenciar melhor as atualizações, reinicializações e outras operações. Você pode criar uma configuração de distribuição usando uma taxa de distribuição estática ou uma taxa de distribuição exponencial. Para especificar o número máximo de destinos de trabalho a serem informados por minuto, use uma taxa de implantação estática.

Para obter exemplos de definição de taxas de implantação e para obter mais informações sobre a configuração de implantações de trabalhos, consulteConfiguração para implantar e anular trabalhos.

Abort

Você pode criar um conjunto de condições para cancelar implantações quando critérios que você especificou forem atendidos. Para obter mais informações, consulteConfiguração para implantar e anular trabalhos.

Tempo limite atingido

Os tempos limite de Job notificam você sempre que uma implantação de trabalho fica presa na`IN_PROGRESS`estado por um período de tempo inesperadamente longo. Há dois tipos de temporizadores: em andamento e de etapa. Quando o trabalho é`IN_PROGRESS`Você pode monitorar e acompanhar o andamento da implantação do trabalho.

As configurações de implementações e abortamento são específicas do seu trabalho, enquanto a configuração de tempo limite é específica para uma implantação de trabalho. Para obter mais informações, consulte Configurações de tempo limite e repetição de execução do Job (p. 720).

Tentativas

As novas tentativas de Job permitem repetir a execução do trabalho quando um trabalho falha, o tempo limite ou ambos. Você pode ter até dez tentativas de tentativas para executar o trabalho. Você pode monitorar e acompanhar o andamento da tentativa de nova tentativa e se a execução do trabalho foi bem-sucedida.

As configurações de implementações e abortamento são específicas para sua tarefa, enquanto as configurações de tempo limite e repetição são específicas para a execução de uma tarefa. Para obter mais informações, consulte Configurações de tempo limite e repetição de execução do Job (p. 720).

Estados de execução de trabalhos e trabalhos

As seções a seguir descrevem o ciclo de vida de umAWS IoTtrabalho e o ciclo de vida de uma execução de trabalho.

Estados de Job

O diagrama a seguir mostra os diferentes estados de umAWS IoTtrabalho.

Um trabalho que você cria usandoAWS IoTOs trabalhos podem estar em um dos seguintes estados:

- IN_PROGRESS

  Ao criar um trabalho usando oAWS IoTConsole do ou oCreateJobAPI, o status do trabalho muda para`IN_PROGRESS`. Todos os trabalhos que estão nesse estado contam para o limite de trabalhos ativos.

  Durante a criação de empregos,AWS IoTOs trabalhos começam a lançar execuções de trabalhos em dispositivos em seu grupo-alvo. Um trabalho nesta fase tem `oIsConcurrent`propriedade definida como true e conta para o limite de simultaneidade do trabalho. Depois que todas as execuções de empregos tiverem sido lançadas,AWS IoTOs trabalhos aguardam que os dispositivos concluam a ação remota. Um trabalho nesta fase tem `oIsConcurrent`Propriedade definida como false e não conta para o limite de simultaneidade do trabalho.

  O limite de trabalhos ativos e o limite de simultaneidade do trabalho se aplicam a trabalhos contínuos e instantâneos. Para obter mais informações, consulte Limites de trabalhos (p. 775).
- COMPLETED

  Um trabalho contínuo está sempre em andamento e continua a executar para todos os novos dispositivos que são adicionados ao grupo de destino. Para um trabalho de instantâneo, o status do trabalho muda para`COMPLETED`quando todas as execuções de trabalho entram em um estado terminal, como`SUCCEEDED`,`FAILED`,`TIMED_OUT`,`REMOVED`, ou`CANCELED`.
- CANCELED

  Ao cancelar um trabalho usando oAWS IoTConsole do, o"CancelJob"API ou oConfiguração de anulação de Job (p. 719), o status do trabalho muda para`CANCELED`.

  Durante o cancelamento do trabalho,AWS IoTOs trabalhos começam a cancelar as execuções de tarefas criadas anteriormente. Um trabalho nesta fase tem `oIsConcurrent`propriedade definida como true e conta para o limite de simultaneidade do trabalho. Depois que todas as execuções de emprego forem canceladas, `oIsConcurrent`propriedade é definida como false e o job não conta para o limite de simultaneidade do trabalho.

O limite se aplica a trabalhos contínuos e instantâneos. Para obter mais informações, consulte Limites de trabalhos (p. 775).

- DELETION_IN_PROGRESS

  Quando você exclui um trabalho usando oAWS IoTConsole do ou oDeleteJobAPI, o status do trabalho muda para`DELETION_IN_PROGRESS`. Durante a exclusão do trabalho,AWS IoTOs trabalhos começam a excluir execuções de tarefas criadas anteriormente. Depois de todas as execuções de trabalho terem sido excluídas, o trabalho desaparece do seuAWSconta.

Estados de execução de Job

O diagrama a seguir mostra os diferentes estados de umAWS IoTexecução de trabalho.

Uma execução de trabalho que é implementada quando você cria um trabalho comAWS IoTOs trabalhos podem estar em um dos seguintes estados:

- QUEUED

  QuandoAWS IoTOs trabalhos implementam uma execução de trabalho para um dispositivo de destino, o status da execução do trabalho é definido como`QUEUED`. A execução do trabalho permanece no`QUEUED`state até:

  - Seus dispositivos recebem a execução do trabalho e invocam as APIs de Jobs e relata o status como`IN_PROGRESS`.
  - Você cancela a execução do job ou do job, ou quando os critérios de anulação especificados usando uma configuração de cancelamento forem atendidos e o status muda para`CANCELED`.
  - Seu dispositivo é removido do grupo de destino e o status muda para`REMOVED`.
- IN_PROGRESS

Se o seu dispositivo IoT se inscrever no reservadoTópicos de trabalhos (p. 109) `$notify`e`$notify-next`, e seu dispositivo invoca o`StartPendingJobExecution`API ou o comando`UpdateJobExecution`API com status de`IN_PROGRESS`,AWS IoTOs trabalhos definirão o status de execução do trabalho como`IN_PROGRESS`. O`UpdateJobExecution`A API pode ser chamada várias vezes com um status de`IN_PROGRESS` e detalhes adicionais sobre as etapas de execução usando o`statusDetails`objeto.

> **Note**
>
> Se você criar vários trabalhos para cada dispositivo,AWS IoTOs trabalhos e o protocolo MQTT não garantem a ordem de entrega.

- SUCCEEDED

  Quando o dispositivo concluir com êxito a operação remota, o dispositivo deve invocar o`UpdateJobExecution`API com status de`SUCCEEDED`para indicar que a execução do trabalho foi bem-sucedida.AWS IoT Os trabalhos atualizam e retornam o status de execução do trabalho como`SUCCEEDED`. Este é um status de terminal.

- FAILED (COM FALHA)

  Quando o dispositivo não concluir a operação remota, o dispositivo deve invocar o`UpdateJobExecution`API com status de`Failed`para indicar que a execução do trabalho falhou.AWS IoT Os trabalhos atualizam e retornam o status de execução do trabalho como`Failed`. Este é um status de terminal. Você pode repetir essa execução de trabalho para o dispositivo usando oConfiguração de nova tentativa de execução de Job (p. 721).

- TIMED_OUT

  Quando o dispositivo não concluir a operação remota dentro do tempo limite do temporizador em andamento,AWS IoTJobs define o status da execução do trabalho como`TIMED_OUT`. Esta duração do temporizador é especificada usando o`inProgressTimeoutInMinutes`propriedade doConfiguração de tempo limite de execução do Job (p. 720). Este é um status de terminal. Você pode repetir essa execução de trabalho para o dispositivo usando oConfiguração de nova tentativa de execução de Job (p. 721).

- REJEITADA

  Quando o dispositivo recebe uma solicitação inválida ou incompatível, o dispositivo deve invocar o`UpdateJobExecution`API com status de`REJECTED`.AWS IoT Os trabalhos atualizam e retornam o status de execução do trabalho como`REJECTED`. Este é um status de terminal. Você não pode repetir essa execução de trabalho para o dispositivo.

- REMOVIDO

  Quando o dispositivo não é mais um destino válido para a execução do trabalho, como quando ele é desanexado de um grupo de coisas dinâmico,AWS IoTJobs define o status da execução do trabalho como`REMOVED`. Você pode reanexar a coisa ao seu grupo-alvo e reiniciar a execução do trabalho para o dispositivo.

- CANCELED

  Quando você cancela uma tarefa ou cancela uma execução de trabalho usando o console ou o`CancelJob`ou`CancelJobExecution`API, ou quando os critérios de abortamento especificados usando oConfiguração de anulação de Job (p. 719)é atendido,AWS IoTJobs cancela o trabalho e define o status de execução do trabalho como`CANCELED`.

Gerenciar trabalhos

Use trabalhos para notificar dispositivos de atualização de firmware ou software. Você pode usar oAWS IoTconsole, oOperações de API de gerenciamento e controle de Job (p. 745), oAWS Command Line Interface, ou oAWSSDKs daPara criar e gerenciar trabalhos.

Assinatura de código para trabalhos

Ao enviar código para dispositivos, para que os dispositivos detectem se o código foi modificado em trânsito, recomendamos que você assine o arquivo de código usando oAWS CLI. Para obter instruções, consulte ??? (p. 691).

Para obter mais informações, consultePara que serve a assinatura de códigoAWS IoT?.

documento de Job

Antes de criar um trabalho, você deve criar um documento de trabalho. Se vocêestá usando assinatura de código paraAWS IoT, você deve carregar seu documento de trabalho em um bucket do Amazon S3 com versionamento. Para obter mais informações sobre como criar um bucket do Amazon S3 e o upload de arquivos para ele, consulteConceitos básicos do Amazon Simple Storage ServicenoGuia de conceitos básicos do Amazon S3.

> **Tip**
>
> Para obter exemplos de documentos de trabalho, consulte o exemplo jobs-agent.js no AWS IoT SDK para JavaScript.

Pre-signed URLs

Seu documento de trabalho pode conter um URL pré-assinado do Amazon S3 que aponta para o arquivo de código (ou outro arquivo). Presigned URLs do Amazon S3 são válidos apenas por um período limitado e são gerados quando um dispositivo solicita um documento de trabalho. Em vez disso, o URL pré-signed não é criado ao criar o documento de trabalho, use um URL de espaço reservado no seu documento de trabalho. Um URL de espaço reservado se parece com o seguinte:

```
${aws:iot:s3-presigned-url:https://s3.region.amazonaws.com/<bucket>/<code
file>}
```

em que:

- *balde*É o bucket do Amazon S3 que contém o arquivo de código.
- *arquivo de código*É a chave do Amazon S3 do arquivo de código.

Quando um dispositivo solicita o documento de trabalho, a AWS IoT gera o presigned URL e substitui o URL de espaço reservado pelo pre-signed URL. Seu documento de trabalho é, então, enviado para o dispositivo.

Função do IAM para conceder permissão para baixar arquivos do S3

Ao criar um trabalho que usa pre-signed URLs do Amazon S3, você deve fornecer uma função do IAM que concede permissão para fazer download de arquivos do bucket do Amazon S3 onde os dados ou as atualizações estão armazenados. A função também deve conceder permissão para a AWS IoT assumir a função.

Você pode especificar um limite de tempo opcional para o pre-signed URL. Para obter mais informações, consulteCreateJob.

GrantAWS IoTPermissão de trabalhos para assumir sua função

1. Vá para oHub de funções do console do IAME selecione sua função.
2. NoRelações de confiança, selecioneEditar relação de confiançaE substitua o documento de política pelo JSON a seguir. Escolha Update Trust Policy.

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

3. Para se proteger contra o problema confuso do deputado, adicione as chaves de contexto da condição globalaws:SourceArneaws:SourceAccountpara a política.

   Important

   Suasaws:SourceArnDeve estar em conformidade com o formato: arn:aws:iot:*region*:*account-id*:*. Certifique-se de que*região*corresponde ao seuAWS IoTRegião e*account-id*corresponde ao ID da sua conta de cliente. Para obter mais informações, consultePrevenção contra o ataque “Confused deputy” em todos os serviços (p. 342).

```
{
  "Effect": "Allow",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service":
          "iot.amazonaws.com"
       },
      "Action": "sts:AssumeRole",
      "Condition": {
         "StringEquals": {
            "aws:SourceAccount": "123456789012"
         },
         "ArnLike": {
              "aws:SourceArn": "arn:aws:iot:*:123456789012:job/*"
         }
       }
     }
   ]
}
```

4. Se o trabalho usar um documento de trabalho que seja um objeto do Amazon S3, escolhaPermissõesE com o JSON a seguir, adicione uma política que conceda permissão para fazer download de arquivos do bucket do Amazon S3:

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

```
        {
            "Effect": "Allow",
            "Action": "s3:GetObject",
            "Resource": "arn:aws:s3:::your_S3_bucket/*"
        }
    ]
}
```

Tópicos

- Crie e gerencie trabalhos usando oAWS Management Console (p. 690)
- Crie e gerencie trabalhos usando oAWS CLI (p. 691)

Crie e gerencie trabalhos usando oAWS Management Console

Como criar um trabalho

1. Escolha seu tipo de trabalho

   1. Vá para oHub de Job doAWS IoTconsolee escolhaCriar trabalho.
   2. Dependendo do dispositivo que você estiver usando, você pode criar um trabalho personalizado, um trabalho de atualização FreeRTOS OTA ou umAWS IoT Greengrasstrabalho. Para este exemplo, selecioneCriar um trabalho personalizado.

2. Insira propriedades do trabalho

   Insira um nome de trabalho alfanumérico exclusivo, descrição opcional e tags e, em seguida, escolhaPróximo.

   > **Note**
   >
   > É recomendável não usar informações pessoalmente identificáveis em IDs de trabalhos e descrição.

3. Escolha seus alvos

   Escolha como alvo de trabalho as coisas ou grupos de coisas que você deseja executar neste trabalho.

4. Especifique seu documento de trabalho

   Você pode carregar o arquivo de trabalho JSON para um bucket do S3 e, em seguida, usar esse arquivo como documento de trabalho ou escolher o arquivo de trabalho em um modelo.

   Se você estiver usando um modelo, você pode escolher entre um modelo de trabalho personalizado ou umAWSModelo gerenciado. Se você estiver criando um trabalho para executar ações remotas usadas com freqüência, como reiniciar o dispositivo, você pode usar umAWSModelo gerenciado. Esses modelos já foram pré-configurados para uso. Para obter mais informações, consulte Criar um modelo de trabalho personalizado (p. 713) e Criar modelos de trabalho personalizados a partir de modelos gerenciados (p. 709).

5. Escolha seu tipo de trabalho

   NoConfiguração do Job, escolha o tipo de trabalho comocontínuoou um trabalho de snapshot. Um trabalho de instantâneo é concluído quando ele termina sua execução nos dispositivos e grupos de destino. Um trabalho contínuo se aplica a grupos de coisas e é executado em qualquer dispositivo que você adiciona posteriormente a um grupo-alvo especificado.

6. Especifique configurações adicionais (opcional)

   Continue adicionando configurações adicionais para seu trabalho e, em seguida, revise e crie seu trabalho. Para obter mais informações sobre configurações adicionais, consulte:

   - Configurações para implantar e anular Job (p. 718)
   - Configurações de tempo limite e repetição de execução do Job (p. 720)

Depois de criar o trabalho, o console gera uma assinatura JSON e a coloca no seu documento de trabalho. Você pode usar o console da AWS IoT para visualizar o status, cancelar ou excluir um trabalho. Para gerenciar trabalhos, vá para oHub de Job do console.

Crie e gerencie trabalhos usando oAWS CLI

Esta seção descreve como criar e gerenciar trabalhos.

Criar trabalhos

Para criar umAWS IoTtrabalho, use oCreateJobcomando. O trabalho é enfileirado para execução nos destinos (coisas ou grupos de coisas) que você especifica. Para criar umAWS IoTTrabalho, você precisa de um documento de trabalho que possa ser incluído no corpo da solicitação ou como um link para um documento do Amazon S3. Se o trabalho incluir arquivos de download que usam pre-signed URLs do Amazon S3, você precisará de uma função do IAM Amazon Resource Name (ARN) do que tenha permissão para fazer download do arquivo e conceda permissão para oAWS IoTServiço de trabalho para assumir a função.

Assinatura de código com trabalhos

Se você estiver usando assinatura de código paraAWS IoT, você deve iniciar um trabalho de assinatura de código e incluir a saída em seu documento de trabalho. Usar atrabalho de início de assinaturaPara criar um trabalho de assinatura de código.`start-signing-job`O retorna um ID de trabalho. Para obter o local do Amazon S3 em que a assinatura é armazenada, use adescribe-signing-jobcomando. Em seguida, você pode fazer download da assinatura do Amazon S3. Para obter mais informações sobre trabalhos de assinatura de código, consulteAssinatura de código paraAWS IoT.

Seu documento de trabalho deve conter um espaço reservado de pre-signed URL para o arquivo de código e a saída da assinatura JSON inserida em um bucket do Amazon S3 usando ostart-signing-jobcomando, incluído em um`codesign`Elemento:

```
{
    "presign": "${aws:iot:s3-presigned-url:https://s3.region.amazonaws.com/bucket/image}",
    "codesign": {
        "rawPayloadSize": <image-file-size>,
        "signature": <signature>,
        "signatureAlgorithm": <signature-algorithm>,
        "payloadLocation": {
            "s3": {
                "bucketName": <my-s3-bucket>,
                "key": <my-code-file>,
                "version": <code-file-version-id>
            }
        }
    }
}
```

Criar um trabalho com um documento de trabalho

O comando a seguir mostra como criar um trabalho usando um documento de trabalho (*job-document.json*) armazenado em um bucket do Amazon S3 (*jobBucket*) e uma função com permissão para fazer download de arquivos do Amazon S3 (*S3DownloadRole*).

```
aws iot create-job  \
      --job-id 010  \
      --targets arn:aws:iot:us-east-1:123456789012:thing/thingOne  \
      --document-source https://s3.amazonaws.com/my-s3-bucket/job-document.json  \
      --timeout-config inProgressTimeoutInMinutes=100 \
      --job-executions-rollout-config "{ \"exponentialRate\": { \"baseRatePerMinute\": 50,
 \"incrementFactor\": 2, \"rateIncreaseCriteria\": { \"numberOfNotifiedThings\": 1000,
 \"numberOfSucceededThings\": 1000}}, \"maximumPerMinute\": 1000}" \
      --abort-config "{ \"criteriaList\": [ { \"action\": \"CANCEL\", \"failureType\":
 \"FAILED\", \"minNumberOfExecutedThings\": 100, \"thresholdPercentage\": 20}, { \"action
\": \"CANCEL\", \"failureType\": \"TIMED_OUT\", \"minNumberOfExecutedThings\": 200,
 \"thresholdPercentage\": 50}]}" \
      --presigned-url-config "{\"roleArn\":\"arn:aws:iam::123456789012:role/
S3DownloadRole\", \"expiresInSec\":3600}"
```

O trabalho é executado em*ThingOne*.

O parâmetro `timeout-config` opcional especifica o tempo que cada dispositivo tem para concluir a execução do trabalho. O temporizador é iniciado quando o status da execução do trabalho é definido como `IN_PROGRESS`. Se o status da execução do trabalho não estiver definido como outro estado terminal antes que o tempo expire, ele será definido como`TIMED_OUT`.

O temporizador em andamento não pode ser atualizado e é aplicado a todas as execuções do trabalho. Sempre que uma execução de trabalho permanece no`IN_PROGRESS`O estado por mais tempo que esse intervalo, ele falhará e alternará para o terminal`TIMED_OUT`status.AWS IoTtambém publica uma notificação MQTT.

Para obter mais informações sobre como criar configurações para implantar e anular trabalhos, consulteConfiguração para Implantar e Anular Trabalhos.

> **Note**
>
> Os documentos de Job que são especificados como arquivos do Amazon S3 são recuperados no momento em que você cria o trabalho. Se você alterar o conteúdo do arquivo do Amazon S3 que usou como a origem do seu documento de trabalho depois de ter criado o documento de trabalho, o que é enviado para os destinos de trabalho não mudará.

Atualizar um trabalho

Para atualizar um trabalho, use oUpdateJobcomando. Você pode atualizar os campos `description`, `presignedUrlConfig`, `jobExecutionsRolloutConfig`, `abortConfig` e `timeoutConfig` de um trabalho.

```
aws iot update-job  \
  --job-id 010  \
  --description "updated description" \
  --timeout-config inProgressTimeoutInMinutes=100 \
  --job-executions-rollout-config "{ \"exponentialRate\": { \"baseRatePerMinute\": 50,
 \"incrementFactor\": 2, \"rateIncreaseCriteria\": { \"numberOfNotifiedThings\": 1000,
 \"numberOfSucceededThings\": 1000}, \"maximumPerMinute\": 1000}}" \
  --abort-config "{ \"criteriaList\": [ { \"action\": \"CANCEL\", \"failureType\":
 \"FAILED\", \"minNumberOfExecutedThings\": 100, \"thresholdPercentage\": 20}, { \"action
\": \"CANCEL\", \"failureType\": \"TIMED_OUT\", \"minNumberOfExecutedThings\": 200,
 \"thresholdPercentage\": 50}]}" \
```

```
  --presigned-url-config "{\"roleArn\":\"arn:aws:iam::123456789012:role/S3DownloadRole\",
 \"expiresInSec\":3600}"
```

Para obter mais informações, consulte Configuração de distribuição e anulação de trabalhos.

Cancelar um trabalho

Para cancelar um trabalho, você usa oCancelJobcomando. O cancelamento de um trabalho fará com que a AWS IoT pare de distribuir novas execuções do trabalho. Ele também cancela qualquer execução de emprego que esteja em um`QUEUED`estado.AWS IoTMantém as execuções de trabalho em estado final inalteradas porque o dispositivo já concluiu o trabalho. Se o status de uma execução de trabalho for `IN_PROGRESS`, ele também permanecerá inalterado, a menos que você use o parâmetro opcional `--force`.

O comando a seguir mostra como cancelar um trabalho com ID 010.

```
aws iot cancel-job --job-id 010
```

O comando exibe a seguinte saída:

```
{
    "jobArn": "string",
    "jobId": "string",
    "description": "string"
}
```

Ao cancelar um trabalho, as execuções de trabalho em estado `QUEUED` serão canceladas. Execuções de Job que estão em um`IN_PROGRESS`O estado são cancelados, mas somente se você especificar o opcional`--force`parâmetro . As execuções de Job em estado final não são canceladas.

> **Warning**
>
> Cancelamento de um trabalho que está no`IN_PROGRESS`state (definindo o`--force`O parâmetro) cancela as execuções de trabalho em andamento e impedirá que o dispositivo que está executando o trabalho atualize o status da execução do trabalho. Tenha cuidado eCertifique-se de que todo dispositivo que esteja executando um trabalho cancelado possa ser recuperado para um estado válido.

O status de um trabalho cancelado ou de uma de suas execuções em algum momento torna-se consistente. A AWS IoT interrompe o mais rápido possível a programação de novas execuções `QUEUED` e não apresenta execuções desse trabalho aos dispositivos. A alteração do status de uma execução de trabalho para `CANCELED` pode levar algum tempo, dependendo do número de dispositivos e de outros fatores.

Se um trabalho for cancelado porque atende aos critérios definidos por um objeto `AbortConfig`, o serviço adicionará valores de preenchimento automático para os campos `reasonCode` e `comment`. Você pode criar seus próprios valores para`reasonCode` Quando o cancelamento de trabalho for acionado pelo usuário.

Cancelar uma execução de trabalho

Para cancelar uma execução de trabalho em um dispositivo, use aCancelJobExecutioncomando. Ele cancelará uma execução de trabalho que esteja em um`QUEUED`estado. Se desejar cancelar uma execução de trabalho em andamento, deverá usar a`--force`parâmetro .

O comando a seguir mostra como cancelar a execução de trabalho 010 em execução em `myThing`.

```
aws iot cancel-job-execution --job-id 010 --thing-name myThing
```

O comando não exibe nenhuma saída.

Execução de trabalho que esteja em um`QUEUED`O estado está cancelado. Execução de trabalho que esteja em um`IN_PROGRESS`state é cancelado, mas somente se você especificar o opcional`--force`parâmetro . As execuções de Job em estado final não podem ser canceladas.

> **Warning**
>
> Cancelar uma execução de trabalho no`IN_PROGRESS`state, o dispositivo não pode atualizar o status da execução do trabalho. Tenha cuidado eCertifique-se de queQue o dispositivo possa ser recuperado para um estado válido.

Se a execução do trabalho estiver em estado final ou se a execução do trabalho estiver em um`IN_PROGRESS`O estado e o`--force`O parâmetro não está definido como`true`, esse comando causa um`InvalidStateTransitionException`.

O status de uma execução de trabalho cancelada é eventualmente consistente. Alterar o status de uma execução de trabalho para`CANCELED`Pode levar algum tempo, dependendo de vários fatores.

Excluir um trabalho

Para excluir um trabalho e suas execuções de trabalho, use aDeleteJobcomando. Por padrão, você só pode excluir um trabalho em estado terminal (`SUCCEEDED`ou`CANCELED`). Caso contrário, ocorrerá uma exceção. Você pode excluir um trabalho no`IN_PROGRESS`state, no entanto, se o`force`O parâmetro está definido como`true`.

Para excluir um trabalho, execute o seguinte comando:

```
aws iot delete-job --job-id 010 --force|--no-force
```

O comando não exibe nenhuma saída.

> **Warning**
>
> Quando você exclui um trabalho que está no`IN_PROGRESS`O dispositivo que está implantando o trabalho não pode acessar informações do trabalho nem atualizar o status da execução do trabalho. Tenha cuidado e certifique-se de que todo dispositivo que esteja implantando um trabalho que tenha sido excluído possa ser recuperado para um estado válido.

A exclusão de um trabalho pode levar algum tempo, dependendo do número de execuções criadas para o trabalho e de outros fatores. Enquanto o trabalho está sendo excluído, `DELETION_IN_PROGRESS` aparece como status do trabalho. Ocorrerá erro se você tentar excluir ou cancelar um trabalho com status já`DELETION_IN_PROGRESS`.

Somente 10 trabalhos podem ter, simultaneamente, o status `DELETION_IN_PROGRESS`. Caso contrário, um `LimitExceededException` ocorre.

Obter um documento de trabalho

Para recuperar um documento de trabalho para um trabalho, use aGetJobDocumentcomando. Um documento de trabalho é uma descrição das operações remotas a serem realizadas pelos dispositivos.

Para obter um documento de trabalho, execute o seguinte comando:

```
aws iot get-job-document --job-id 010
```

O comando retorna o documento de trabalho para o trabalho especificado:

```
{
    "document": "{\n\t\"operation\":\"install\",\n\t\"url\":\"http://amazon.com/
firmWareUpate-01\",\n\t\"data\":\"${aws:iot:s3-presigned-url:https://s3.amazonaws.com/job-
test-bucket/datafile}\"\n}"
}
```

> **Note**
>
> Ao usar esse comando para recuperar um documento de trabalho, URLs de espaço reservado são substituídas por pre-signed URLs do Amazon S3. Quando um dispositivo chama oGetPendingJobExecutionsEm operação da API, URLs de espaço reservado são substituídas por pre-signed URLs do Amazon S3 no documento de trabalho.

Listar trabalhos

Para obter uma lista de todos os trabalhos em seuConta da AWS, use oListJobscomando. Os dados do trabalho e os dados de execução do trabalho são retidos por um tempo limitado. Execute o comando a seguir para listar todas as tarefas em seuConta da AWS:

```
aws iot list-jobs
```

O comando retorna todos os trabalhos em sua conta, classificados pelo status do trabalho:

```
{
    "jobs": [
        {
            "status": "IN_PROGRESS",
            "lastUpdatedAt": 1486687079.743,
            "jobArn": "arn:aws:iot:us-east-1:123456789012:job/013",
            "createdAt": 1486687079.743,
            "targetSelection": "SNAPSHOT",
            "jobId": "013"
        },
        {
            "status": "SUCCEEDED",
            "lastUpdatedAt": 1486685868.444,
            "jobArn": "arn:aws:iot:us-east-1:123456789012:job/012",
            "createdAt": 1486685868.444,
            "completedAt": 148668789.690,
            "targetSelection": "SNAPSHOT",
            "jobId": "012"
        },
        {
            "status": "CANCELED",
            "lastUpdatedAt": 1486678850.575,
            "jobArn": "arn:aws:iot:us-east-1:123456789012:job/011",
            "createdAt": 1486678850.575,
            "targetSelection": "SNAPSHOT",
            "jobId": "011"
        }
    ]
}
```

Descrever um trabalho

Para obter o status de um trabalho, execute oDescribeJobcomando. O comando a seguir mostra como descrever um trabalho:

```
$ aws iot describe-job --job-id 010
```

O comando retorna o status do trabalho especificado. Por exemplo:

```
{
    "documentSource": "https://s3.amazonaws.com/job-test-bucket/job-document.json",
    "job": {
        "status": "IN_PROGRESS",
        "jobArn": "arn:aws:iot:us-east-1:123456789012:job/010",
        "targets": [
            "arn:aws:iot:us-east-1:123456789012:thing/myThing"
        ],
        "jobProcessDetails": {
            "numberOfCanceledThings": 0,
            "numberOfFailedThings": 0,
            "numberOfInProgressThings": 0,
            "numberOfQueuedThings": 0,
            "numberOfRejectedThings": 0,
            "numberOfRemovedThings": 0,
            "numberOfSucceededThings": 0,
            "numberOfTimedOutThings": 0,
            "processingTargets": [
                arn:aws:iot:us-east-1:123456789012:thing/thingOne,
                arn:aws:iot:us-east-1:123456789012:thinggroup/thinggroupOne,
                arn:aws:iot:us-east-1:123456789012:thing/thingTwo,
                arn:aws:iot:us-east-1:123456789012:thinggroup/thinggroupTwo
            ]
        },
        "presignedUrlConfig": {
            "expiresInSec": 60,
            "roleArn": "arn:aws:iam::123456789012:role/S3DownloadRole"
        },
        "jobId": "010",
        "lastUpdatedAt": 1486593195.006,
        "createdAt": 1486593195.006,
        "targetSelection": "SNAPSHOT",
        "jobExecutionsRolloutConfig": {
            "exponentialRate": {
                "baseRatePerMinute": integer,
                "incrementFactor": integer,
                "rateIncreaseCriteria": {
                    "numberOfNotifiedThings": integer, // Set one or the other
                    "numberOfSucceededThings": integer // of these two values.
                },
            "maximumPerMinute": integer
         }
        },
        "abortConfig": {
            "criteriaList": [
                {
                    "action": "string",
                    "failureType": "string",
                    "minNumberOfExecutedThings": integer,
                    "thresholdPercentage": integer
                }
            ]
        },
        "timeoutConfig": {
```

```
            "inProgressTimeoutInMinutes": number
           }
    }
}
```

Listar execuções para um trabalho

Um trabalho em execução em um dispositivo específico é representado por um objeto de execução de trabalho. Execute o comando ListJobExecutionsForJob para listar todas as execuções de um trabalho. O exemplo a seguir mostra como listar as execuções de um trabalho:

```
aws iot list-job-executions-for-job --job-id 010
```

O comando retorna uma lista de execuções do trabalho:

```
{
    "executionSummaries": [
    {
        "thingArn": "arn:aws:iot:us-east-1:123456789012:thing/thingOne",
        "jobExecutionSummary": {
            "status": "QUEUED",
            "lastUpdatedAt": 1486593196.378,
            "queuedAt": 1486593196.378,
            "executionNumber": 1234567890
        }
    },
    {
        "thingArn": "arn:aws:iot:us-east-1:123456789012:thing/thingTwo",
        "jobExecutionSummary": {
            "status": "IN_PROGRESS",
            "lastUpdatedAt": 1486593345.659,
            "queuedAt": 1486593196.378,
            "startedAt": 1486593345.659,
            "executionNumber": 4567890123
        }
    }
    ]
}
```

Listar execuções de trabalho para uma coisa

Execute o comando ListJobExecutionsForThing para listar todas as execuções de trabalho em execução em uma coisa. O exemplo a seguir mostra como listar execuções de trabalho para uma coisa:

```
aws iot list-job-executions-for-thing --thing-name thingOne
```

O comando retorna uma lista de execuções de trabalho que estão em execução ou que executaram na coisa especificada:

```
{
    "executionSummaries": [
    {
        "jobExecutionSummary": {
            "status": "QUEUED",
            "lastUpdatedAt": 1486687082.071,
            "queuedAt": 1486687082.071,
            "executionNumber": 9876543210
        },
        "jobId": "013"
```

```
    },
    {
        "jobExecutionSummary": {
            "status": "IN_PROGRESS",
            "startAt": 1486685870.729,
            "lastUpdatedAt": 1486685870.729,
            "queuedAt": 1486685870.729,
            "executionNumber": 1357924680
        },
        "jobId": "012"
    },
    {
        "jobExecutionSummary": {
            "status": "SUCCEEDED",
            "startAt": 1486678853.415,
            "lastUpdatedAt": 1486678853.415,
            "queuedAt": 1486678853.415,
            "executionNumber": 4357680912
        },
        "jobId": "011"
    },
    {
        "jobExecutionSummary": {
            "status": "CANCELED",
            "startAt": 1486593196.378,
            "lastUpdatedAt": 1486593196.378,
            "queuedAt": 1486593196.378,
            "executionNumber": 2143174250
        },
        "jobId": "010"
    }
    ]
}
```

Descrever a execução do trabalho

Execute o comando DescribeJobExecution para obter o status de execução de um trabalho. Você deve especificar um ID de trabalho e um nome de coisa e, opcionalmente, um número de execução para identificar a execução do trabalho. O comando a seguir mostra como descrever a execução de um trabalho:

```
aws iot describe-job-execution --job-id 017 --thing-name thingOne
```

O comando retorna a `JobExecution`. Por exemplo:

```
{
    "execution": {
        "jobId": "017",
        "executionNumber": 4516820379,
        "thingArn": "arn:aws:iot:us-east-1:123456789012:thing/thingOne",
        "versionNumber": 123,
        "createdAt": 1489084805.285,
        "lastUpdatedAt": 1489086279.937,
        "startedAt": 1489086279.937,
        "status": "IN_PROGRESS",
        "approximateSecondsBeforeTimedOut": 100,
        "statusDetails": {
            "status": "IN_PROGRESS",
            "detailsMap": {
                "percentComplete": "10"
            }
        }
```

```
    }
}
```

Excluir uma execução de trabalho

Execute o comando DeleteJobExecution para excluir a execução de um trabalho. Você deve especificar um ID de trabalho, um nome de coisa e um número de execução para identificar a execução do trabalho. O comando a seguir mostra como excluir a execução de um trabalho:

```
aws iot delete-job-execution --job-id 017 --thing-name thingOne --execution-number
 1234567890 --force|--no-force
```

O comando não exibe nenhuma saída.

Por padrão, o status de execução do trabalho deve ser `QUEUED` ou em estado terminal (`SUCCEEDED`, `FAILED`, `REJECTED`, `TIMED_OUT`, `REMOVED` ou `CANCELED`). Caso contrário, ocorrerá um erro. Para excluir uma execução de trabalho com status `IN_PROGRESS`, você pode definir o parâmetro `force` como `true`.

> **Warning**
>
> Ao excluir uma execução de trabalho com status `IN_PROGRESS`, o dispositivo que está executando o trabalho não pode acessar informações do trabalho nem atualizar o status da execução do trabalho. Tenha cuidado e certifique-se de que o dispositivo possa ser recuperado para um estado válido.

Templates de trabalho

Use modelos de trabalho para pré-configurar trabalhos que você pode implantar em vários conjuntos de dispositivos de destino. Para ações remotas executadas com frequência que você deseja implantar em seus dispositivos, como reinicializar ou instalar um aplicativo, você pode usar modelos para definir configurações padrão. Quando você quiser executar operações como a implantação de patches de segurança e correções de bugs, você também pode criar modelos a partir de trabalhos existentes.

Ao criar um modelo de trabalho, você pode especificar as seguintes configurações e recursos adicionais.

- Propriedades do Job
- Documentos e destinos de Job
- Critérios de implantação e cancelamento
- Critérios de tempo limite

Personalizado eAWSModelos gerenciados

Dependendo da ação remota que você deseja executar, você pode criar um modelo de trabalho personalizado ou usar umAWSModelo gerenciado. Use modelos de trabalho personalizados para fornecer seu próprio documento de trabalho personalizado e criar trabalhos reutilizáveis para implantar em seus dispositivos.AWSmodelos gerenciados são modelos de trabalho fornecidos porAWS IoTTrabalhos para ações comumente executadas. Esses modelos têm um documento de trabalho predefinido para algumas ações remotas, para que você não precise criar seu próprio documento de trabalho. Modelos gerenciados ajudam você a criar trabalhos reutilizáveis para execução mais rápida em seus dispositivos.

Tópicos

- Usar oAWSmodelos gerenciados para implantar operações remotas comuns (p. 700)
- Criar modelos de trabalho personalizados (p. 712)

Usar oAWSmodelos gerenciados para implantar operações remotas comuns

AWSmodelos gerenciados são modelos de trabalho fornecidos porAWSpara ações remotas executadas com freqüência, como reinicialização, download de um arquivo ou instalação de um aplicativo em seus dispositivos. Esses modelos têm um documento de trabalho predefinido para cada ação remota para que você não precise criar seu próprio documento de trabalho.

Você pode escolher entre um conjunto de configurações predefinidas e criar trabalhos usando esses modelos sem escrever nenhum código adicional. Usando modelos gerenciados, você pode visualizar o documento de trabalho implantado em suas frotas. Você pode criar um trabalho usando esses modelos e criar ainda mais um modelo de trabalho personalizado que você pode reutilizar para suas operações remotas.

O que contêm modelos gerenciados do?

CadaAWSmodelo gerenciado contém:

- Um documento de trabalho que especifica o nome da operação e seus parâmetros. Por exemplo, se você usar umdownload filemodelo, o nome da operação édownload filee os parâmetros podem ser:
  - O URL do arquivo que você deseja baixar para o dispositivo, que pode ser um recurso da Internet ou um URL do S3 público ou pré-assinado.
  - Um caminho de arquivo local no dispositivo para armazenar o arquivo baixado.

  Para obter mais informações sobre os documentos de trabalhos e seus parâmetros, consulteAções remotas de modelo gerenciado e documentos de trabalho (p. 701).
- O ambiente para executar os comandos no documento de trabalho.

Pré-requisitos

Para que seus dispositivos executem as ações remotas especificadas pelo documento de trabalho de modelo gerenciado, você deve:

- Instale o software específico em seu dispositivo
  - Usar aAWS IoTCliente do dispositivo

    Recomendamos instalar e executar aAWS IoTDevice Client em seus dispositivos porque ele suporta o uso de todos os modelos gerenciados diretamente do console por padrão.

    O Device Client é um software de código aberto escrito em C++ que você pode compilar e instalar em seus dispositivos IoT baseados em Linux incorporados. O Device Client tem umcliente basee discretorecursos do lado do cliente do. O cliente base estabelece conectividade comAWS IoTatravés do protocolo MQTT e pode se conectar com os diferentes recursos do lado do cliente.

    Para executar operações remotas em seus dispositivos, use oFuncionalidade de trabalhos no lado do cliente dodo cliente do dispositivo. Esse recurso contém um analisador para receber o documento de trabalho e os manipuladores de tarefas que implementam as ações remotas especificadas no documento de trabalho. Para obter mais informações sobre o Device Client e seus recursos, consulteAWS IoTCliente do dispositivo.

    Ao executar em dispositivos, o Device Client recebe o documento de trabalho e tem uma implementação específica da plataforma que ele usa para executar comandos no documento. Para obter mais informações sobre como configurar o Device Client e usar o recurso Trabalhos, consulteTutoriais do AWS IoT (p. 125).

- Use seu próprio software de dispositivo e manipuladores de trabalho

  Alternativamente, você pode escrever seu próprio código para os dispositivos usando aAWS IoTO SDK do dispositivo e sua biblioteca de manipuladores que suportam as operações remotas. Para implantar e executar trabalhos, certifique-se de que as bibliotecas do agente de dispositivo foram instaladas corretamente e estejam em execução nesses dispositivos.

  Você pode ainda optar usar seus próprios manipuladores que podem suportar as operações remotas. Para obter informações sobre como você pode criar esses manipuladores, consulteManipuladores de trabalho de amostra noAWS IoTRepositório GitHub do cliente do dispositivo.
- Usar um ambiente compatível

  Para cada modelo gerenciado, você encontrará informações sobre o ambiente que você pode usar para executar as ações remotas. Recomendamos usar o modelo com um ambiente Linux compatível, conforme especificado no modelo. Você pode usar oAWS IoTDevice Client para executar as ações remotas do modelo de gerenciamento porque suporta microprocessadores comuns e ambientes Linux, como Debian e Ubuntu.

Ações remotas de modelo gerenciado e documentos de trabalho

A seção a seguir lista os diferentesAWSModelos gerenciados doAWS IoTTrabalhos e descreve as ações remotas que podem ser executadas nos dispositivos. Na seção abaixo, você encontrará informações sobre o documento de trabalho e uma descrição dos parâmetros do documento de trabalho para cada ação remota. O software do lado do dispositivo usa o nome do modelo e os parâmetros para executar a ação remota.

AWSos modelos gerenciados aceitam parâmetros de entrada para os quais você especifica um valor ao criar um trabalho usando o modelo. Todos os modelos gerenciados têm dois parâmetros de entrada opcionais em comum:`runAsUser`e`pathToHandler`. Except`AWS-Reboot`, os modelos requerem parâmetros de entrada adicionais para os quais você deve especificar um valor ao criar um trabalho usando o modelo. Esses parâmetros de entrada necessários variam de acordo com o modelo escolhido. Por exemplo, se você escolher o`AWS-Download-File`modelo, você deve especificar uma lista de pacotes para instalar e um URL para baixar arquivos.

Você especifica um valor para os parâmetros de entrada ao usar aAWS IoTConsole ou oAWS CLIPara criar um trabalho que usa um modelo gerenciado. Ao usar a CLI, você fornece esses valores usando a`document-parameters`objeto. Para obter mais informações, consultedocumentParameters.

> **Note**
>
> Você deve usar`document-parameters`somente ao criar trabalhos a partir deAWSmodelos gerenciados. Esse parâmetro não pode ser usado com modelos de trabalho personalizados ou para criar trabalhos a partir deles.

A seguir, mostra uma descrição dos parâmetros de entrada opcionais comuns. Você verá uma descrição de outros parâmetros de entrada que cada modelo gerenciado requer na seção abaixo.

`runAsUser`

Esse parâmetro especifica se o manipulador de trabalho será executado como outro usuário. Se não for especificado durante a criação do trabalho, o manipulador de trabalho será executado como o mesmo usuário que o Device Client. Quando você executar o manipulador de trabalho como outro usuário, especifique um valor de string que não seja superior a 256 caracteres.

`pathToHandler`

O caminho para o manipulador de trabalho em execução no dispositivo. Se não for especificado durante a criação do trabalho, o Device Client usará o diretório de trabalho atual.

A seguir, mostra as diferentes ações remotas, seus documentos de trabalho e os parâmetros que eles aceitam. Todos esses modelos suportam o ambiente Linux para executar a operação remota no dispositivo.

AWS—Reinicializar

Nome do modelo

`AWS-Reboot`

Descrição do modelo

Um modelo gerenciado fornecido porAWSpara reiniciar o dispositivo.

Parâmetros de entrada

Esse modelo não possui parâmetros obrigatórios. Você pode especificar os parâmetros opcionais`runAsUser`e`pathToHandler`.

comportamento do dispositivo

O dispositivo é reinicializado com sucesso.

documento de Job

A seguir, mostra o documento de trabalho e sua versão mais recente. O modelo mostra o caminho para o manipulador de trabalho e o script shell,`reboot.sh`, que o manipulador de trabalho deve ser executado para reinicializar o dispositivo.

```
{
  "version": "1.0",
  "steps": [
    {
      "action": {
        "name": "Reboot",
        "type": "runHandler",
        "input": {
          "handler": "reboot.sh",
          "path": "${aws:iot:parameter:pathToHandler}"
        },
        "runAsUser": "${aws:iot:parameter:runAsUser}"
      }
    }
  ]
}
```

AWS—Download — Arquivo

Nome do modelo

`AWS-Download-File`

Descrição do modelo

Um modelo gerenciado fornecido porAWSPara baixar um arquivo.

Parâmetros de entrada

Esse modelo tem os seguintes parâmetros obrigatórios. Você também pode especificar os parâmetros opcionais`runAsUser`e`pathToHandler`.

`downloadUrl`

A URL para baixar o arquivo, que pode ser um recurso da Internet, um objeto no Amazon S3 que pode ser acessado publicamente ou um objeto no Amazon S3 que só pode ser acessado pelo dispositivo usando um URL pré-assinado. Para obter mais informações sobre como usar URLs pré-assinados e conceder permissões, consultePre-signed URLs (p. 688).

`filePath`

Um caminho de arquivo local que mostra a localização no dispositivo para armazenar o arquivo baixado.

comportamento do dispositivo

O dispositivo baixa o arquivo do local especificado, verifica se o download está concluído e o armazena localmente.

documento de Job

A seguir, mostra o documento de trabalho e sua versão mais recente. O modelo mostra o caminho para o manipulador de trabalho e o script shell,`download-file.sh`, que o manipulador de trabalho deve ser executado para baixar o arquivo. Ela também mostra os parâmetros necessários`downloadUrlfilePath`.

```
{
  "version": "1.0",
  "steps": [
    {
      "action": {
        "name": "Download-File",
        "type": "runHandler",
        "input": {
          "handler": "download-file.sh",
          "args": [
            "${aws:iot:parameter:downloadUrl}",
            "${aws:iot:parameter:filePath}"
          ],
          "path": "${aws:iot:parameter:pathToHandler}"
        },
        "runAsUser": "${aws:iot:parameter:runAsUser}"
      }
    }
  ]
}
```

AWS—Instalar — Aplicativo

Nome do modelo

`AWS-Install-Application`

Descrição do modelo

Um modelo gerenciado fornecido porAWSPara instalar um ou mais aplicativos.

Parâmetros de entrada

Esse modelo tem o parâmetro necessário a seguir,`packages`. Você também pode especificar os parâmetros opcionais`runAsUserpathToHandler`.

`packages`

Uma lista separada por espaços de um ou mais aplicativos a serem instalados.

comportamento do dispositivo

O dispositivo instala os aplicativos conforme especificado no documento de trabalho.

documento de Job

A seguir, mostra o documento de trabalho e sua versão mais recente. O modelo mostra o caminho para o manipulador de trabalho e o script shell,`install-packages.sh`, que o manipulador de trabalho deve ser executado para baixar o arquivo. Ela também mostra o parâmetro necessário`packages`.

```
{
  "version": "1.0",
  "steps": [
    {
      "action": {
        "name": "Install-Application",
        "type": "runHandler",
        "input": {
          "handler": "install-packages.sh",
          "args": [
            "${aws:iot:parameter:packages}"
          ],
          "path": "${aws:iot:parameter:pathToHandler}"
        },
        "runAsUser": "${aws:iot:parameter:runAsUser}"
      }
    }
  ]
}
```

AWS—Remover — Aplicativo

Nome do modelo

`AWS-Remove-Application`

Descrição do modelo

Um modelo gerenciado fornecido porAWSPara desinstalar um ou mais aplicativos.

Parâmetros de entrada

Esse modelo tem o parâmetro necessário a seguir,`packages`. Você também pode especificar os parâmetros opcionais`runAsUser`e`pathToHandler`.

`packages`

Uma lista separada por espaços de um ou mais aplicativos a serem desinstalados.

comportamento do dispositivo

O dispositivo desinstala os aplicativos conforme especificado no documento de trabalho.

documento de Job

A seguir, mostra o documento de trabalho e sua versão mais recente. O modelo mostra o caminho para o manipulador de trabalho e o script shell,`remove-packages.sh`, que o manipulador de trabalho deve ser executado para baixar o arquivo. Ela também mostra o parâmetro necessário`packages`.

```
{
  "version": "1.0",
  "steps": [
    {
```

```
      "action": {
        "name": "Remove-Application",
        "type": "runHandler",
        "input": {
          "handler": "remove-packages.sh",
          "args": [
            "${aws:iot:parameter:packages}"
          ],
          "path": "${aws:iot:parameter:pathToHandler}"
        },
        "runAsUser": "${aws:iot:parameter:runAsUser}"
      }
    }
  ]
}
```

AWS—Start — Aplicativo

Nome do modelo

`AWS-Start-Application`

Descrição do modelo

Um modelo gerenciado fornecido porAWSPara iniciar um ou mais serviços.

Parâmetros de entrada

Esse modelo tem o parâmetro necessário a seguir,`services`. Você também pode especificar os parâmetros opcionais`runAsUser`e`pathToHandler`.

`services`

Uma lista separada por espaços de um ou mais aplicativos a serem iniciados.

comportamento do dispositivo

Os aplicativos especificados começam a ser executados no dispositivo.

documento de Job

A seguir, mostra o documento de trabalho e sua versão mais recente. O modelo mostra o caminho para o manipulador de trabalho e o script shell,`start-services.sh`, que o manipulador de trabalho deve ser executado para iniciar os serviços do sistema. Ela também mostra o parâmetro necessário`services`.

```
{
  "version": "1.0",
  "steps": [
    {
      "action": {
        "name": "Start-Application",
        "type": "runHandler",
        "input": {
          "handler": "start-services.sh",
          "args": [
            "${aws:iot:parameter:services}"
          ],
          "path": "${aws:iot:parameter:pathToHandler}"
        },
        "runAsUser": "${aws:iot:parameter:runAsUser}"
      }
    }
  ]
}
```

AWS—Stop — Aplicativo

Nome do modelo

`AWS-Stop-Application`

Descrição do modelo

Um modelo gerenciado fornecido porAWSPara interromper um ou mais serviços.

Parâmetros de entrada

Esse modelo tem o parâmetro necessário a seguir,`services`. Você também pode especificar os parâmetros opcionais`runAsUser`e`pathToHandler`.

`services`

Uma lista separada por espaços de um ou mais aplicativos a serem interrompidos.

comportamento do dispositivo

Os aplicativos especificados param de ser executados no dispositivo.

documento de Job

A seguir, mostra o documento de trabalho e sua versão mais recente. O modelo mostra o caminho para o manipulador de trabalho e o script shell,`stop-services.sh`, que o manipulador de trabalho deve ser executado para interromper os serviços do sistema. Ela também mostra o parâmetro necessário`services`.

```
{
  "version": "1.0",
  "steps": [
    {
      "action": {
        "name": "Stop-Application",
        "type": "runHandler",
        "input": {
          "handler": "stop-services.sh",
          "args": [
            "${aws:iot:parameter:services}"
          ],
          "path": "${aws:iot:parameter:pathToHandler}"
        },
        "runAsUser": "${aws:iot:parameter:runAsUser}"
      }
    }
  ]
}
```

AWS—Reiniciar — Aplicativo

Nome do modelo

`AWS-Restart-Application`

Descrição do modelo

Um modelo gerenciado fornecido porAWSpara parar e reiniciar um ou mais serviços.

Parâmetros de entrada

Esse modelo tem o parâmetro necessário a seguir,`services`. Você também pode especificar os parâmetros opcionais`runAsUser`e`pathToHandler`.

`services`

Uma lista separada por espaços de um ou mais aplicativos a serem reiniciados.

comportamento do dispositivo

Os aplicativos especificados são interrompidos e reiniciados no dispositivo.

documento de Job

A seguir, mostra o documento de trabalho e sua versão mais recente. O modelo mostra o caminho para o manipulador de trabalho e o script shell,`restart-services.sh`, que o manipulador de trabalho deve ser executado para reiniciar os serviços do sistema. Ela também mostra o parâmetro necessário`services`.

```
{
  "version": "1.0",
  "steps": [
    {
      "action": {
        "name": "Restart-Application",
        "type": "runHandler",
        "input": {
          "handler": "restart-services.sh",
          "args": [
            "${aws:iot:parameter:services}"
          ],
          "path": "${aws:iot:parameter:pathToHandler}"
        },
        "runAsUser": "${aws:iot:parameter:runAsUser}"
      }
    }
  ]
}
```

Tópicos

- Criar um trabalho a partir deAWSmodelos gerenciados usando oAWS Management Console (p. 707)
- Criar um trabalho doAWSmodelos gerenciados usando oAWS CLI (p. 709)

Criar um trabalho a partir deAWSmodelos gerenciados usando oAWS Management Console

Usar aAWS Management ConsolePara obter informações sobreAWSmodelos gerenciados e crie um trabalho usando esses modelos. Você pode salvar o trabalho criado como seu próprio modelo personalizado.

Obter detalhes sobre modelos gerenciados

Você pode obter informações sobre os diferentes modelos gerenciados que estão disponíveis para uso noAWS IoTconsole do .

1. Para ver seus modelos gerenciados disponíveis, acesse oHub de modelos de Job doAWS IoTconsolee escolha oModelos gerenciadosGuia.
2. Escolha um modelo gerenciado para ver os detalhes.

A página de detalhes contém as seguintes informações:

- Nome, descrição e Nome de recurso da Amazon (ARN) do modelo gerenciado.

- O ambiente no qual as operações remotas podem ser executadas, como o Linux.
- O documento de trabalho JSON que especifica o caminho para o manipulador de tarefas e os comandos a serem executados no dispositivo. Por exemplo, o seguinte mostra um exemplo de documento de trabalho para oReinicializar AWS. O modelo mostra o caminho para o manipulador de trabalho e o script shell,`reboot.sh`, que o manipulador de trabalho deve ser executado para reinicializar o dispositivo.

```
{
  "version": "1.0",
  "steps": [
    {
      "action": {
        "name": "Reboot",
        "type": "runHandler",
        "input": {
          "handler": "reboot.sh",
          "path": "${aws:iot:parameter:pathToHandler}"
        },
        "runAsUser": "${aws:iot:parameter:runAsUser}"
      }
    }
  ]
}
```

  Para obter mais informações sobre o documento de trabalho e seus parâmetros para várias ações remotas, consulteAções remotas de modelo gerenciado e documentos de trabalho (p. 701).
- A versão mais recente do documento de trabalho.

Criar um trabalho usando modelos gerenciados

Você pode usar o console do para escolher umAWSModelo gerenciado para usar para criar um trabalho. Esta seção explica como fazer isso.

Você também pode iniciar o fluxo de trabalho de criação de trabalho e, em seguida, escolher oAWSModelo gerenciado que você deseja usar ao criar o trabalho. Para obter mais informações sobre esse fluxo de trabalho, consulteCrie e gerencie trabalhos usando oAWS Management Console (p. 690).

1. Escolha oAWSModelo gerenciado

   Vá para oHub de modelos de Job doAWS IoTconsole, escolha oModelos gerenciadosE, em seguida, escolha o modelo.
2. Crie um trabalho usando seu modelo gerenciado

   1. Na página de detalhes do modelo, escolhaCriar trabalho.

      O console muda para oPropriedades de trabalho personalizadasetapa doCriar trabalhoO fluxo de trabalho em que a configuração do modelo foi adicionada.
   2. Insira um nome de trabalho alfanumérico exclusivo, descrição e tags opcionais e, em seguida, escolhaPróximo.
   3. Escolha as coisas ou grupos de coisas como destinos de trabalho que você deseja executar neste trabalho.
   4. Nodocumento de Job, seu modelo é exibido com suas configurações e parâmetros de entrada. Insira valores para os parâmetros de entrada do modelo escolhido. Por exemplo, se você escolher oAWS-Baixar arquivo: Modelo
      - para oBaixar URL, digite a URL do arquivo para download, por exemplo:`https://`*`example.com/index`*`.html`.
      - para ofilePath, insira o caminho no dispositivo para armazenar o arquivo baixado, por exemplo:*`path/to/file`*.

Você também pode inserir valores para os parâmetros`runAsUserepathToHandler`. Para obter mais informações sobre os parâmetros de entrada de cada modelo, consulteAções remotas de modelo gerenciado e documentos de trabalho (p. 701).

5. NoConfiguração do Job, escolha o tipo de trabalho comocontínuoou um trabalho de snapshot. Um trabalho de instantâneo é concluído quando ele termina sua execução nos dispositivos e grupos de destino. Um trabalho contínuo se aplica a grupos de coisas e é executado em qualquer dispositivo que você adicionar a um grupo-alvo especificado.
6. Continue adicionando configurações adicionais para seu trabalho e, em seguida, revise e crie seu trabalho. Para obter mais informações sobre configurações adicionais, consulte:
   - Configurações para implantar e anular Job (p. 718)
   - Configurações de tempo limite e repetição de execução do Job (p. 720)

Criar modelos de trabalho personalizados a partir de modelos gerenciados

Você pode usar umAWSmodelo gerenciado e um trabalho personalizado como ponto de partida para criar seu próprio modelo de trabalho personalizado. Para criar um modelo de trabalho personalizado, primeiro crie um trabalho do seu serviço doAWSModelo gerenciado, conforme descrito na seção anterior.

Em seguida, você pode salvar o trabalho personalizado como um modelo para criar seu próprio modelo de trabalho personalizado. Para salvar como modelo:

1. Vá para oHub de Job doAWS IoTconsolee escolha o trabalho que contém seu modelo gerenciado.
2. SelecioneSalvar como modelo de trabalhoEm seguida, crie seu modelo de trabalho personalizado. Para obter mais informações sobre como criar um modelo de trabalho personalizado, consulteCrie um modelo de trabalho com base em um trabalho existente (p. 713).

Criar um trabalho doAWSmodelos gerenciados usando oAWS CLI

Usar aAWS CLIPara obter informações sobreAWSmodelos gerenciados e crie um trabalho usando esses modelos. Em seguida, você pode salvar o trabalho como modelo e, em seguida, criar seu próprio modelo personalizado.

Listar modelos gerenciados

O`list-managed-job-templates` AWS CLIO comando relaciona todos os modelos de trabalho em seuConta da AWS.

```
aws iot list-managed-job-templates
```

Por padrão, a execução deste comando exibe todos os disponíveisAWSmodelos gerenciados e seus detalhes.

```
{
    "managedJobTemplates": [
        {
            "templateArn": "arn:aws:iot:region::jobtemplate/AWS-Reboot:1.0",
            "templateName": "AWS-Reboot",
            "description": "A managed job template for rebooting the device.",
            "environments": [
                "LINUX"
```

```
            ],
            "templateVersion": "1.0"
        },
        {
            "templateArn": "arn:aws:iot:region::jobtemplate/AWS-Remove-Application:1.0",
            "templateName": "AWS-Remove-Application",
            "description": "A managed job template for uninstalling one or more
 applications.",
            "environments": [
                "LINUX"
            ],
            "templateVersion": "1.0"
        },
        {
            "templateArn": "arn:aws:iot:region::jobtemplate/AWS-Stop-Application:1.0",
            "templateName": "AWS-Stop-Application",
            "description": "A managed job template for stopping one or more system
 services.",
            "environments": [
                "LINUX"
            ],
            "templateVersion": "1.0"
        },

        ...

        {
            "templateArn": "arn:aws:iot:us-east-1::jobtemplate/AWS-Restart-
Application:1.0",
            "templateName": "AWS-Restart-Application",
            "description": "A managed job template for restarting one or more system
 services.",
            "environments": [
                "LINUX"
            ],
            "templateVersion": "1.0"
        }
    ]
}
```

Para obter mais informações, consulteListManagedJobTemplates.

Obter detalhes sobre um modelo gerenciado

O`describe-managed-job-template` AWS CLIO comando obtém detalhes sobre um modelo de trabalho especificado. Especifique o nome do modelo de trabalho e uma versão de modelo opcional. Se a versão do modelo não for especificada, a versão padrão predefinida será retornada. A seguir, mostra um exemplo de execução do comando para obter detalhes sobre o`AWS-Download-File`.

```
aws iot describe-managed-job-template \
    --template-name AWS-Download-File
```

O comando exibe os detalhes do modelo e o ARN, seu documento de trabalho e o`documentParameters`, que é uma lista de pares de valores-chave de parâmetros de entrada do modelo. Para obter informações sobre os diferentes modelos e parâmetros de entrada, consulteAções remotas de modelo gerenciado e documentos de trabalho (p. 701).

> **Note**
>
> O`documentParameters`objeto retornado quando você usa essa API só deve ser usado ao criar trabalhos deAWSmodelos gerenciados. O objeto não deve ser usado para modelos de trabalho

personalizados. Para obter um exemplo que mostra como usar esse parâmetro, consulteCrie um trabalho usando modelos gerenciados (p. 711).

```
{
    "templateName": "AWS-Download-File",
    "templateArn": "arn:aws:iot:region::jobtemplate/AWS-Download-File:1.0",
    "description": "A managed job template for downloading a file.",
    "templateVersion": "1.0",
    "environments": [
        "LINUX"
    ],
    "documentParameters": [
        {
            "key": "downloadUrl",
            "description": "URL of file to download.",
            "regex": "(.*?)",
            "example": "http://www.example.com/index.html",
            "optional": false
        },
        {
            "key": "filePath",
            "description": "Path on the device where downloaded file is written.",
            "regex": "(.*?)",
            "example": "/path/to/file",
            "optional": false
        },
        {
            "key": "runAsUser",
            "description": "Execute handler as another user. If not specified, then handler
 is executed as the same user as device client.",
            "regex": "(.){0,256}",
            "example": "user1",
            "optional": true
        },
        {
            "key": "pathToHandler",
            "description": "Path to handler on the device. If not specified, then device
 client will use the current working directory.",
            "regex": "(.){0,4096}",
            "example": "/path/to/handler/script",
            "optional": true
        }
    ],
    "document": "{\"version\":\"1.0\",\"steps\":[{\"action\":{\"name
\":\"Download-File\",\"type\":\"runHandler\",\"input\":{\"handler\":
\"download-file.sh\",\"args\":[\"${aws:iot:parameter:downloadUrl}\",
\"${aws:iot:parameter:filePath}\"],\"path\":\"${aws:iot:parameter:pathToHandler}\"},
\"runAsUser\":\"${aws:iot:parameter:runAsUser}\"}}]}"
}
```

Para obter mais informações, consulteDescribeManagedJobTemplate.

Crie um trabalho usando modelos gerenciados

O`create-job` AWS CLIO comando pode ser usado para criar um trabalho com base em um modelo de trabalho. Ele tem como alvo um dispositivo chamado`thingOne`E especifica o nome de recurso da Amazon (ARN) do modelo gerenciado a ser usado como base do trabalho. Você pode substituir configurações avançadas, como configurações de tempo limite e cancelamento, passando os parâmetros associados do`create-job`comando.

O exemplo mostra como criar um trabalho que usa a`AWS-Download-File`. Ele também mostra como especificar os parâmetros de entrada do modelo usando o`document-parameters`parâmetro .

> **Note**
>
> Você deve usar `odocument-parameters`objeto somente comAWSmodelos gerenciados. Esse objeto não deve ser usado com modelos de trabalho personalizados.

```
aws iot create-job \
   --targets arn:aws:iot:region:account-id:thing/thingOne \
   --job-id "new-managed-template-job" \
   --job-template-arn arn:aws:iot:region::jobtemplate/AWS-Download-File:1.0 \
   --document-parameters downloadUrl=https://example.com/index.html,filePath=path/to/file
```

em que:

- *região*é oRegião da AWS.
- *account-id*é o únicoConta da AWSnúmero.
- *thingOne*é o nome da coisa da IoT para a qual o trabalho é direcionado.
- *AWS-Download do arquivo: 1.0*É o nome do modelo gerenciado.
- `https://`*example.com/index*`.html`É o URL do qual fazer download do arquivo.
- `https://`*pathto/file/index*é o caminho no dispositivo para armazenar o arquivo baixado.

Executar este comando cria um trabalho para o modelo,*AWS-Download-File*, como mostrado abaixo.

```
{
    "jobArn": "arn:aws:iot:region:account-id:job/new-managed-template-job",
    "jobId": "new-managed-template-job",
    "description": "A managed job template for downloading a file."
}
```

Crie um modelo de trabalho personalizado com base em modelos gerenciados

1. Crie um trabalho usando um modelo gerenciado, conforme descrito na seção anterior.
2. Crie um modelo de trabalho personalizado usando o ARN do trabalho que você criou. Para obter mais informações, consulte Crie um modelo de trabalho com base em um trabalho existente (p. 716).

Criar modelos de trabalho personalizados

Você pode criar modelos de trabalho usando oAWS CLIO e aAWS IoTconsole do . Você também pode criar trabalhos a partir de modelos de trabalho usando oAWS CLI, oAWS IoTconsole e Fleet Hub forAWS IoTAplicativos da Web de gerenciamento de dispositivos. Para obter mais informações sobre como trabalhar com modelos de trabalhos em aplicativos Fleet Hub, consulteTrabalhando com modelos de trabalho no Fleet Hub paraAWS IoTGerenciamento de dispositivos.

Tópicos

- Crie modelos de trabalho personalizados usando oAWS Management Console (p. 712)
- Crie modelos de trabalho personalizados usando oAWS CLI (p. 715)

Crie modelos de trabalho personalizados usando oAWS Management Console

Este tópico explica como criar, excluir e exibir detalhes sobre modelos de trabalhos usando aAWS IoTconsole do .

Criar um modelo de trabalho personalizado

Você também pode criarUm originalmodelo de trabalho personalizado ou crie um modelo de trabalho a partir de um trabalho existente. Você pode ainda criar um modelo de trabalho personalizado a partir de um trabalho existente que foi criado usando um modelo de trabalho personalizadoAWSModelo gerenciado. Para obter mais informações, consulte Criar modelos de trabalho personalizados a partir de modelos gerenciados (p. 709).

Criar um modelo de trabalho original

1. Comece a criar seu modelo de trabalho

   1. Vá para oHub de modelos de Job doAWS IoTconsolee escolha oModelos personalizadosGuia.
   2. SelecioneCriar modelo de trabalho.

      Note

      Você também pode navegar até oTemplates de trabalhopágina doServiços relacionadospágina abaixoHub de frotas.

2. Especificar propriedades do modelo de trabalho

   NoCriar modelo de trabalho, insira um identificador alfanumérico para o nome do trabalho e uma descrição alfanumérica para fornecer detalhes adicionais sobre o modelo.

   Note

   Não recomendamos o uso de informações pessoalmente identificáveis em IDs de trabalhos ou descrições.

3. Fornecer documento de trabalho

   Forneça um arquivo de trabalho JSON armazenado em um bucket do S3 ou como um documento de trabalho embutido especificado dentro da tarefa. Esse arquivo de trabalho se tornará o documento de trabalho quando você criar um trabalho usando esse modelo.

   Se o arquivo de trabalho estiver armazenado em um bucket S3, insira a URL do S3 ou escolhaNavegar no S3e, em seguida, navegue até o documento de trabalho e selecione-o.

   Note

   Você pode selecionar apenas buckets do S3 em sua região atual.

4. Continue adicionando configurações adicionais para seu trabalho e, em seguida, revise e crie seu trabalho. Para obter mais informações sobre configurações adicionais, consulte:

   - Configurações para implantar e anular Job (p. 718)
   - Configurações de tempo limite e repetição de execução do Job (p. 720)

Crie um modelo de trabalho com base em um trabalho existente

1. Escolha seu trabalho

   1. Vá para oHub de Job doAWS IoTconsoleE escolha o trabalho que você deseja usar como base para seu modelo de trabalho.
   2. SelecioneSalvar como modelo de trabalho.

> **Note**
>
> Opcionalmente, você pode escolher um documento de trabalho diferente ou editar as configurações avançadas do trabalho original e, em seguida, escolherCriar modelo de trabalho. Seu novo modelo de trabalho é exibido noTemplates de trabalho.

2. Especificar propriedades do modelo de trabalho

   NoCriar modelo de trabalho, insira um identificador alfanumérico para o nome do trabalho e uma descrição alfanumérica para fornecer detalhes adicionais sobre o modelo.

   > **Note**
   >
   > O documento de trabalho é o arquivo de trabalho que você especificou ao criar o modelo. Se o documento de trabalho for especificado dentro do trabalho em vez de um local do S3, você poderá ver o documento de trabalho na página de detalhes desse trabalho.

3. Continue adicionando configurações adicionais para seu trabalho e, em seguida, revise e crie seu trabalho. Para obter mais informações sobre configurações adicionais, consulte:

   - Configurações para implantar e anular Job (p. 718)
   - Configurações de tempo limite e repetição de execução do Job (p. 720)

Crie um trabalho com base em um modelo de trabalho personalizado

Você pode criar um trabalho a partir de um modelo de trabalho personalizado acessando a página de detalhes do modelo de trabalho, conforme descrito neste tópico. Você também pode criar um trabalho ou escolhendo o modelo de trabalho que deseja usar ao executar o fluxo de trabalho de criação de trabalho. Para obter mais informações, consulte Crie e gerencie trabalhos usando oAWS Management Console (p. 690).

Este tópico mostra como criar um trabalho da página de detalhes de um modelo de trabalho personalizado. Você também pode criar um trabalho a partir de umAWSModelo gerenciado. Para obter mais informações, consulte Criar um trabalho usando modelos gerenciados (p. 708).

1. Escolha o modelo de trabalho personalizado

   Vá para oHub de modelos de Job doAWS IoTconsolee escolha oModelos personalizadosE, em seguida, escolha o modelo.

2. Crie um trabalho usando seu modelo personalizado

   Para criar um trabalho:

   1. Na página de detalhes do modelo, escolhaCriar trabalho.

      O console muda para oPropriedades de trabalho personalizadasetapa doCriar trabalhoO fluxo de trabalho em que a configuração do modelo foi adicionada.
   2. Insira um nome de trabalho alfanumérico exclusivo, descrição e tags opcionais e, em seguida, escolhaPróximo.
   3. Escolha as coisas ou grupos de coisas como destinos de trabalho que você deseja executar neste trabalho.

      Nodocumento de Job, seu modelo é exibido com suas configurações. Se você quiser usar outro documento de trabalho, escolhaNavegare selecione um intervalo e um documento diferentes. Escolha Next (Próximo).
   4. NoConfiguração do Job, escolha o tipo de trabalho comocontínuoou um trabalho de snapshot. Um trabalho de instantâneo é concluído quando ele termina sua execução nos dispositivos e grupos de

destino. Um trabalho contínuo se aplica a grupos de coisas e é executado em qualquer dispositivo que você adicionar a um grupo-alvo especificado.

5. Continue adicionando configurações adicionais para seu trabalho e, em seguida, revise e crie seu trabalho. Para obter mais informações sobre configurações adicionais, consulte:
   - Configurações para implantar e anular Job (p. 718)
   - Configurações de tempo limite e repetição de execução do Job (p. 720)

Você também pode criar trabalhos a partir de modelos de trabalho com aplicativos da Web do Fleet Hub. Para obter informações sobre como criar trabalhos no Fleet Hub, consulteTrabalhando com modelos de trabalho no Fleet Hub paraAWS IoTGerenciamento de dispositivos.

Excluir um modelo de trabalho

Para excluir um modelo de trabalho, primeiro vá para aHub de modelos de Job doAWS IoTconsolee escolha oModelos personalizadosGuia. Em seguida, escolha o modelo de trabalho que você deseja excluir e escolhaExcluir.

> **Note**
>
> Uma exclusão é permanente e o modelo de trabalho não aparece mais noModelos personalizadosGuia.

Crie modelos de trabalho personalizados usando oAWS CLI

Este tópico explica como criar, excluir e recuperar detalhes sobre modelos de trabalhos usando aAWS CLI.

Criar um modelo de trabalho do zero

Os seguintes exemplos deAWS CLIO comando mostra como criar um trabalho usando um documento de trabalho (*`job-document.json`*) armazenado em um bucket do S3 e uma função com permissão para baixar arquivos do Amazon S3 (*`S3DownloadRole`*).

```
aws iot create-job-template  \
      --job-template-id 010  \
      --document-source https://s3.amazonaws.com/my-s3-bucket/job-document.json  \
      --timeout-config inProgressTimeoutInMinutes=100 \
      --job-executions-rollout-config "{ \"exponentialRate\": { \"baseRatePerMinute\":
 50, \"incrementFactor\": 2, \"rateIncreaseCriteria\": { \"numberOfNotifiedThings\": 1000,
 \"numberOfSucceededThings\": 1000}}, \"maximumPerMinute\": 1000}" \
      --abort-config "{ \"criteriaList\": [ { \"action\": \"CANCEL\", \"failureType\":
 \"FAILED\", \"minNumberOfExecutedThings\": 100, \"thresholdPercentage\": 20}, { \"action
\": \"CANCEL\", \"failureType\": \"TIMED_OUT\", \"minNumberOfExecutedThings\": 200,
 \"thresholdPercentage\": 50}]}" \
      --presigned-url-config "{\"roleArn\":\"arn:aws:iam::123456789012:role/S3DownloadRole
\", \"expiresInSec\":3600}"
```

O parâmetro `timeout-config` opcional especifica o tempo que cada dispositivo tem para concluir a execução do trabalho. O temporizador é iniciado quando o status da execução do trabalho é definido como `IN_PROGRESS`. Se o status da execução do trabalho não estiver definido como outro estado final antes que o tempo expire, ele será definido como`TIMED_OUT`.

O temporizador em andamento não pode ser atualizado e é aplicado a todas as execuções do trabalho. Sempre que uma execução de trabalho permanece no estado `IN_PROGRESS` por mais tempo que esse intervalo, a execução do trabalho falha e alterna para o status `TIMED_OUT` terminal. O AWS IoT também publica uma notificação MQTT.

Para obter mais informações sobre como criar configurações para implantar e anular trabalhos, consulteConfiguração para implantar e anular trabalhos.

> **Note**
>
> Os documentos de Job que são especificados como arquivos do Amazon S3 são recuperados no momento em que você cria o trabalho. Se você alterar o conteúdo do arquivo do Amazon S3 usado como a origem de seu documento de trabalho depois de criar o trabalho, o que é enviado para os destinos do trabalho não será alterado.

Crie um modelo de trabalho com base em um trabalho existente

Os seguintes exemplos deAWS CLIO comando cria um modelo de trabalho especificando o nome de recurso da Amazon (ARN) de um trabalho existente. O novo modelo de trabalho usa todas as configurações especificadas na tarefa. Opcionalmente, você pode alterar qualquer uma das configurações na tarefa existente usando qualquer um dos parâmetros opcionais.

```
aws iot create-job-template  \
      --job-arn arn:aws:iot:region:123456789012:job/job-name  \
      --timeout-config inProgressTimeoutInMinutes=100
```

Obter detalhes sobre um modelo de trabalho

Os seguintes exemplos deAWS CLIO comando obtém detalhes sobre um modelo de trabalho especificado.

```
aws iot describe-job-template \
      --job-template-id template-id
```

O comando exibe a saída a seguir.

```
{
   "abortConfig": {
      "criteriaList": [
         {
            "action": "string",
            "failureType": "string",
            "minNumberOfExecutedThings": number,
            "thresholdPercentage": number
         }
      ]
   },
   "createdAt": number,
   "description": "string",
   "document": "string",
   "documentSource": "string",
   "jobExecutionsRolloutConfig": {
      "exponentialRate": {
         "baseRatePerMinute": number,
         "incrementFactor": number,
         "rateIncreaseCriteria": {
            "numberOfNotifiedThings": number,
            "numberOfSucceededThings": number
         }
      },
      "maximumPerMinute": number
```

```
   },
   "jobTemplateArn": "string",
   "jobTemplateId": "string",
   "presignedUrlConfig": {
      "expiresInSec": number,
      "roleArn": "string"
   },
   "timeoutConfig": {
      "inProgressTimeoutInMinutes": number
   }
}
```

Listar modelos de trabalho

Os seguintes exemplos deAWS CLIO comando relaciona todos os modelos de trabalho em seuConta da AWS.

```
aws iot list-job-templates
```

O comando exibe a saída a seguir.

```
{
   "jobTemplates": [
      {
         "createdAt": number,
         "description": "string",
         "jobTemplateArn": "string",
         "jobTemplateId": "string"
      }
   ],
   "nextToken": "string"
}
```

Para recuperar páginas adicionais de resultados, use o valor do`nextToken`campo.

Excluir um modelo de trabalho

Os seguintes exemplos deAWS CLIO comando exclui um modelo de trabalho especificado.

```
aws iot delete-job-template \
      --job-template-id template-id
```

O comando não exibe nenhuma saída.

Crie um trabalho com base em um modelo de trabalho personalizado

Os seguintes exemplos deAWS CLIO comando cria um trabalho com base em um modelo de trabalho personalizado. Ele tem como alvo um dispositivo chamado`thingOne`E especifica o nome de recurso da Amazon (ARN) do modelo de trabalho a ser usado como base do trabalho. Você pode substituir configurações avançadas, como configurações de tempo limite e cancelamento, passando os parâmetros associados do`create-job`comando.

Warning

O`document-parameters`objeto deve ser usado com o`create-job`comando somente ao criar trabalhos a partirAWSmodelos gerenciados. Esse objeto não deve ser usado com modelos de trabalho personalizados. Para obter um exemplo que mostra como criar trabalhos usando esse parâmetro, consulteCrie um trabalho usando modelos gerenciados (p. 711).

```
aws iot create-job \
      --targets arn:aws:iot:region:123456789012:thing/thingOne  \
      --job-template-arn arn:aws:iot:region:123456789012:jobtemplate/template-id
```

Trabalhoconfigurações

Você pode ter as seguintes configurações adicionais para cada trabalho implantado nos destinos especificados.

- Distribuição: Essa configuração define quantos dispositivos recebem o documento de trabalho a cada minuto.
- Abort: Use essa configuração para cancelar um trabalho em casos como quando alguns dispositivos não recebem a notificação de trabalho ou seus dispositivos relatam falha para suas execuções de trabalho.
- Timeout (Tempo limite): Se não houver uma resposta de seus objetivos de trabalho dentro de um determinado período após o início das execuções do trabalho, o trabalho poderá falhar.
- Tentar novamente: Se o dispositivo relatar falha ao tentar concluir uma execução de trabalho ou se a execução do trabalho expirar, você poderá usar essa configuração para tentar novamente a execução do trabalho para o dispositivo.

Usando essas configurações, você pode monitorar o status da execução do trabalho e evitar que uma atualização incorreta seja enviada para uma frota inteira.

Tópicos

- Como as configurações de trabalho funcionam (p. 718)
- Especificar configurações adicionais (p. 723)

Como as configurações de trabalho funcionam

Você usa as configurações de distribuição e cancelamento quando estiver implantando um trabalho e as configurações de tempo limite e repetição para execução do trabalho. As seções a seguir mostram mais informações sobre como essas configurações funcionam.

Tópicos

- Configurações para implantar e anular Job (p. 718)
- Configurações de tempo limite e repetição de execução do Job (p. 720)

Configurações para implantar e anular Job

Você pode usar as configurações de distribuição e cancelamento de tarefas para definir quantos dispositivos recebem o documento de trabalho a cada minuto e também os critérios para cancelar uma tarefa quando um determinado número de dispositivos não receber um documento de trabalho.

Configuração de distribuição de Job

Você pode especificar a rapidez com que os destinos são notificados sobre a execução de um trabalho pendente. Você também pode criar uma implantação em etapas para gerenciar melhor as atualizações, reinicializações e outras operações. Para especificar como seus destinos são notificados, use as taxas de distribuição de trabalhos.

Taxas de implantação de Job

Você pode criar uma configuração de distribuição usando uma taxa de distribuição constante ou uma taxa de distribuição exponencial. Para especificar o número máximo de destinos de trabalho a serem informados por minuto, use uma taxa de implantação constante.

AWS IoTOs trabalhos de trabalho podem ser implantados usando taxas de implantação exponencial à medida que vários critérios e limites são atingidos. Se o número de trabalhos com falha corresponder a um conjunto de critérios especificados por você, você poderá cancelar a implantação do trabalho. Você define os critérios de taxa de distribuição do trabalho ao criar um trabalho usando o`JobExecutionsRolloutConfig`objeto. Você também define os critérios de cancelamento da tarefa na criação do trabalho usando o`AbortConfig`objeto.

O exemplo a seguir mostra como as taxas de implantação funcionam. Por exemplo, uma implantação de trabalho com uma taxa base de 50 por minuto, fator de incremento de 2 e número de dispositivos notificados e bem-sucedidos, cada um como 1000, funcionaria da seguinte forma: O trabalho começará a uma taxa de 50 execuções por minuto e continuará a essa taxa até que 1.000 coisas recebessem notificações de execução de trabalho ou 1.000 execuções bem-sucedidas ocorressem.

A tabela a seguir ilustra como a implantação continuaria nos primeiros quatro incrementos.

Taxa de implementação por minuto	50	100	200	400
Número de dispositivos notificados ou execuções bem-sucedidas de trabalho para satisfazer um aumento de taxa	1000	2000	3000	4000

Taxas de distribuição de Job para trabalhos contínuos usando grupos de coisas dinâmicas

Quando você usa um trabalho contínuo para implantar operações remotas em sua frota,AWS IoTJobs implementa execuções de trabalhos para dispositivos em seu grupo de coisas alvo. Para todos os novos dispositivos adicionados ao grupo de coisas dinâmicas, essas execuções de trabalho continuam a ser implementadas para os dispositivos recém-adicionados mesmo após a criação do trabalho.

A configuração de distribuição pode controlar as taxas de distribuição somente para dispositivos que são adicionados ao grupo até a criação do trabalho. Após a criação de um trabalho, para qualquer novo dispositivo, as execuções de tarefas são criadas quase em tempo real assim que os dispositivos se juntam ao grupo-alvo.

Configuração de anulação de Job

Use essa configuração para criar um critério para cancelar um trabalho quando uma porcentagem limite de dispositivos atender a esses critérios. Por exemplo, você pode usar essa configuração para cancelar um trabalho nos seguintes casos:

- Quando uma porcentagem limite de seus dispositivos não recebe as notificações de execução do trabalho, como quando o dispositivo é incompatível com uma atualização OTA. Nesse caso, seu dispositivo pode denunciar um`REJECTED`Status.
- Quando uma porcentagem limite de dispositivos relata falha para suas execuções de trabalho, como quando o dispositivo encontra uma desconexão ao tentar baixar o documento de trabalho de um URL do S3. Nesses casos, seu dispositivo deve ser programado para denunciar o`FAILURE`Status paraAWS IoT.

- Quando um`TIMED_OUT`status é relatado à medida que a execução do trabalho expira para uma porcentagem limite de dispositivos após o início das execuções do trabalho.
- Quando há várias falhas de repetição. Quando você adiciona uma configuração de nova tentativa, cada tentativa de nova tentativa pode incorrer em cobranças adicionais para suaConta da AWS. Nesses casos, cancelar o trabalho pode cancelar as execuções de empregos na fila e evitar tentativas de novas tentativas para essas execuções. Para obter mais informações sobre a configuração de nova tentativa e usá-la com a configuração de anulação, consulteConfigurações de tempo limite e repetição de execução do Job (p. 720).

Você pode configurar uma condição de anulação de trabalho usando aAWS IoTConsole ou oAWS IoTAPI de trabalhos.

Configurações de tempo limite e repetição de execução do Job

Use a configuração de tempo limite de execução do trabalho para enviar vocêNotificações de trabalhos (p. 735)quando uma execução de trabalho estiver em andamento por mais tempo do que a duração definida. Use a configuração de nova tentativa de execução do trabalho para repetir a execução quando o trabalho falhar ou expirar.

Configuração de tempo limite de execução do Job

Use a configuração de tempo limite de execução do trabalho para notificá-lo sempre que uma execução de trabalho ficar presa na`IN_PROGRESS`estado por um período de tempo inesperadamente longo. Quando o trabalho é`IN_PROGRESS`Você pode monitorar o progresso da execução do trabalho.

Temporizadores para tempos limite de trabalho

Há dois tipos de temporizadores: em andamento e de etapa.

Temporizadores em andamento

Ao criar um trabalho ou um modelo de trabalho, você pode especificar um valor para o temporizador em andamento entre 1 minuto e 7 dias. Você pode atualizar o valor desse temporizador até o início da execução da tarefa. Depois de o temporizador começar, ele não poderá ser atualizado e o valor do temporizador é aplicado a todas as execuções do trabalho. Sempre que uma execução de trabalho permanece com o status `IN_PROGRESS` por mais tempo que esse intervalo, a execução falha e alterna para o status `TIMED_OUT` terminal. O AWS IoT também publica uma notificação MQTT.

Temporizador passo

Você também pode definir um temporizador de etapa que se aplica apenas à execução do trabalho que deseja atualizar. Este temporizador não tem efeito sobre o temporizador em andamento. Cada vez que você atualizar uma execução de trabalho, você pode definir um novo valor para o temporizador de etapa. Você também pode criar um novo temporizador de etapa ao iniciar a próxima execução de trabalho pendente para uma coisa. Se a execução de trabalho permanecer com o status `IN_PROGRESS` por mais tempo que o intervalo do temporizador de etapa, ela falhará e alternará para o status `TIMED_OUT` terminal.

> **Note**
>
> Você pode definir o temporizador em andamento usando oAWS IoTConsole ou oAWS IoTAPI de trabalhos. Para especificar o temporizador de etapas, use a API.

Como os temporizadores funcionam para tempos limite de trabalho

O seguinte ilustra como os tempos limite em andamento e de etapa interagem entre si em um período de tempo limite de 20 minutos.

A seguir, mostra as diferentes etapas:

1. 12:00

   Um novo trabalho é criado e um temporizador em andamento por vinte minutos é iniciado ao criar um trabalho. O temporizador em andamento começa a ser executado e a execução do trabalho muda para`IN_PROGRESS`Status.

2. 12:05

   É criado um novo temporizador de etapa com um valor de 7 minutos. A execução do trabalho agora atingirá o tempo limite às 12h12.

3. 1210 PM

   É criado um novo temporizador de etapa com um valor de 5 minutos. Quando um novo temporizador de etapa é criado, o temporizador de etapa anterior é descartado e a execução do trabalho agora atingirá o tempo limite às 12h15.

4. 1213 PM

   É criado um novo temporizador de etapa com um valor de 9 minutos. O temporizador da etapa anterior é descartado e a execução do trabalho agora atingirá o tempo limite às 12:20, porque o temporizador em andamento atinge o tempo limite às 12:20 PM. O temporizador de etapa não pode exceder o limite absoluto do temporizador em andamento.

Configuração de nova tentativa de execução de Job

Você pode usar a configuração de nova tentativa para repetir a execução do trabalho quando um determinado conjunto de critérios for atendido. Uma nova tentativa pode ser tentada quando um trabalho expirar ou quando o dispositivo falhar. Para repetir a execução devido a uma falha de tempo limite, você deve habilitar a configuração de tempo limite.

Como usar a configuração de nova tentativa

Veja a seguir como usar a configuração de nova tentativa:

1. Determine se deve usar a configuração de nova tentativa para`FAILED`,`TIMED_OUT`, ou ambos os critérios de falha. Para o`TIMED_OUT`status, após o status ser relatado,AWS IoTOs trabalhos tentarão automaticamente a execução do trabalho para o dispositivo.
2. Para o`FAILED`status, verifique se a falha na execução do trabalho pode ser repetida. Se for repetido, programe seu dispositivo para denunciar um`FAILURE`Status paraAWS IoT. A seção a seguir descreve mais sobre falhas retentáveis e não retentáveis.
3. Especifique o número de tentativas a serem usadas para cada tipo de falha usando as informações anteriores. Para um único dispositivo, você pode especificar até 10 novas tentativas para ambos os tipos de falha combinados. As tentativas de nova tentativa são interrompidas automaticamente quando uma execução for bem-sucedida ou quando atinge o número especificado de tentativas.
4. Adicione uma configuração de cancelamento para cancelar o trabalho em caso de falhas repetidas de repetição, evitando assim cobranças adicionais no caso de haver um grande número de tentativas de nova tentativa.

Repetir e anular a configuração

Cada tentativa de nova tentativa incorre em cobranças adicionais para oConta da AWS. Para evitar cobranças adicionais de falhas repetidas de repetições, recomendamos adicionar uma configuração de cancelamento. Para obter mais informações sobre a definição de preço, consulte Definição de preço do AWS IoT Device Management.

Você pode encontrar várias falhas de repetição quando uma porcentagem de limite alto de seus dispositivos expirar ou relatar falha. Nesse caso, você pode usar a configuração de cancelamento para cancelar o trabalho e evitar execuções de tarefas na fila ou novas tentativas de repetição.

> **Note**
>
> Quando os critérios de abortamento forem atendidos para cancelar a execução de uma tarefa, somente`QUEUED`execuções de empregos são canceladas. Qualquer nova tentativa em fila para o dispositivo não será tentada. No entanto, as execuções de emprego atuais que têm um`IN_PROGRESS`status não será cancelado.

Antes de tentar novamente uma execução de trabalho com falha, também recomendamos que você verifique se a falha na execução do trabalho é repetida, conforme descrito na seção a seguir.

Tentar novamente o tipo de falha de**`FAILED`**

Para tentar tentativas para o tipo de falha de`FAILED`, seus dispositivos devem ser programados para denunciar o`FAILURE`status para uma execução de trabalho com falhaAWS IoT. Você também deve definir a configuração de nova tentativa com os critérios para tentar novamente`FAILED`execuções de trabalhos e especifique o número de novas tentativas a serem executadas. QuandoAWS IoTOs trabalhos detectam o status FAILURE e, em seguida, tentarão automaticamente tentar novamente a execução da tarefa para o dispositivo. As tentativas continuam até que a execução do trabalho seja bem-sucedida ou quando atinge o número máximo de tentativas de nova tentativa.

Você pode acompanhar cada tentativa de nova tentativa e o trabalho em execução nesses dispositivos. Ao rastrear o status da execução, após a tentativa do número especificado de novas tentativas, você pode usar o dispositivo para relatar falhas e iniciar outra tentativa de nova tentativa.

Falhas retratáveis e não retentáveis

A falha na execução do trabalho pode ser repetida ou não recuperável. Cada tentativa de nova tentativa pode incorrer em cobranças para suaConta da AWS. Para evitar cobranças adicionais de várias tentativas de repetição, primeiro considere verificar se a falha na execução do trabalho é repetida. Um exemplo de falha repetível inclui um erro de conexão que o dispositivo encontra ao tentar baixar o documento de trabalho de um URL do S3. Se a falha na execução do trabalho for repetida, você poderá programar o dispositivo para relatar um`FAILURE`status caso a execução do trabalho falhe e defina a configuração de nova tentativa para tentar novamente`FAILED`execuções.

Se a execução não puder ser repetida, para evitar repetir a execução do trabalho e potencialmente incorrer em cobranças adicionais em sua conta, recomendamos que você programe o dispositivo para denunciar um`REJECTED`Status paraAWS IoT. Exemplos de falha não repetível incluem quando o dispositivo é incompatível de receber uma atualização de trabalho ou quando ele sofre um erro de memória durante a execução de um trabalho. Nestes casos,AWS IoTOs trabalhos não tentarão novamente a execução do trabalho porque ela tenta repetir a execução do trabalho somente quando detecta um`FAILED`ou`TIMED_OUT`Status.

Depois de determinar que uma falha na execução do trabalho pode ser repetida, se uma tentativa de nova tentativa ainda falhar, considere verificar os logs do dispositivo.

Tentar novamente o tipo de falha de**`TIMEOUT`**

Se você habilitar o tempo limite ao criar um trabalho, entãoAWS IoTOs trabalhos tentarão repetir a execução do trabalho para o dispositivo quando o status mudar de`IN_PROGRESS`para`TIMED_OUT`. Essa alteração de status pode ocorrer quando o temporizador em andamento expirar ou quando um temporizador de etapas especificado estiver em`IN_PROGRESS`e, em seguida, esgotar. As tentativas continuam até que a execução do trabalho seja bem-sucedida ou quando atinge o número máximo de tentativas de nova tentativa para esse tipo de falha.

Atualizações contínuas de associação de grupos de coisas e trabalhos

Para trabalhos contínuos que têm status de trabalho como`IN_PROGRESS`, o número de tentativas de nova tentativa é redefinido para zero quando há atualizações para a associação ao grupo de uma coisa. Por exemplo, considere que você especificou cinco tentativas de nova tentativa e três novas tentativas já foram executadas. Se uma coisa agora for removida do grupo de coisas e ela se juntar novamente ao grupo, como no caso de grupos de coisas dinâmicas, o número de tentativas de nova tentativa é redefinido para zero. Agora você pode executar cinco tentativas para o seu grupo de coisas em vez das duas tentativas restantes. Além disso, quando uma coisa é removida do grupo de coisas, tentativas adicionais de novas tentativas são canceladas.

Especificar configurações adicionais

Ao criar um modelo de trabalho ou trabalho, você pode especificar essas configurações adicionais. O seguinte mostra quando você pode especificar essas configurações.

- Ao criar um modelo de trabalho personalizado. As configurações adicionais especificadas serão salvas quando você criar um trabalho a partir do modelo.
- Ao criar um trabalho personalizado usando um arquivo de trabalho. O arquivo de trabalho pode ser um arquivo JSON carregado em um bucket do S3.
- Ao criar um trabalho personalizado usando um modelo de trabalho personalizado. Se o modelo já tiver essas configurações especificadas, você poderá reutilizá-las ou substituí-las especificando novas configurações.
- Ao criar um trabalho personalizado usando umAWSModelo gerenciado.

Tópicos

- Especifique as configurações de tarefa usando oAWS Management Console (p. 723)
- Especifique as configurações de tarefa usando oAWS IoTAPI de trabalhos (p. 725)

Especifique as configurações de tarefa usando oAWS Management Console

Você pode adicionar as diferentes configurações para seu trabalho usando oAWS IoTconsole do . Depois de criar um trabalho, você pode ver os detalhes de status das configurações do trabalho na página

de detalhes do trabalho. Para obter mais informações sobre as diferentes configurações e como elas funcionam, consulteComo as configurações de trabalho funcionam (p. 718).

Adicione as configurações de trabalho ao criar um trabalho ou um modelo de trabalho.

Ao criar um modelo de trabalho personalizado

Para especificar a configuração de distribuição ao criar um modelo de trabalho personalizado

1. Vá para oHub de modelos de Job doAWS IoTconsolee escolhaCriar modelo de trabalho.
2. Especifique as propriedades do modelo de trabalho, forneça o documento de trabalho, expanda a configuração que você deseja adicionar e, em seguida, especifique os parâmetros de configuração.

Ao criar um trabalho personalizado

Para especificar a configuração de distribuição ao criar um trabalho personalizado

1. Vá para oHub de Job doAWS IoTconsolee escolhaCriar trabalho.
2. SelecioneCriar um trabalho personalizadoe especifique as propriedades da tarefa, os destinos e se deseja usar um arquivo de trabalho ou um modelo para o documento de trabalho. Você pode usar um modelo personalizado ou umAWSModelo gerenciado.
3. Escolha a configuração do trabalho e, a seguir, expandaConfiguração de implantaçãoPara especificar se deseja usar umTaxa constanteouTaxa exponencial. Em seguida, especifique os parâmetros de configuração.

A próxima seção mostra os parâmetros que você pode especificar para cada configuração.

Configuração de implantação

Você pode especificar se deve usar uma taxa de distribuição constante ou uma taxa exponencial.

- Definir uma taxa de distribuição constante

  Para definir uma taxa constante para execuções de cargos, escolhaTaxa constantee depois especifique oMáximo por minutoPara o limite superior da taxa. Esse valor é opcional e varia de 1 a 1000. Se você não defini-lo, ele usará 1000 como o valor padrão.
- Definir uma taxa de distribuição exponencial

  Para definir uma taxa exponencial, escolhaTaxa exponenciale, em seguida, especifique esses parâmetros:
  - Taxa base por minuto

    Taxa na qual os trabalhos são executados até aNúmero de dispositivos notificadosouNúmero de dispositivos bem-sucedidoslimite é atingido paraCritérios de aumento de.
  - Fator de incremento

    O fator exponencial em que a taxa de implantação aumenta após aNúmero de dispositivos notificadosouNúmero de dispositivos bem-sucedidoslimite é atingido paraCritérios de aumento de.
  - Critérios de aumento de

    O limite para qualquer umNúmero de dispositivos notificadosouNúmero de dispositivos bem-sucedidos.

Abort configuration

SelecioneAdicionar nova configuraçãoE especifique os seguintes parâmetros para cada configuração:

- Tipo de falha

  Especifica os tipos de falha que iniciam a anulação de um trabalho. Dentre elas estãoFALHARAM,REJEITADA,TIMED_OUT, ouTUDO.
- Fator de incremento

  Especifica o número de execuções de trabalho concluídas que devem ocorrer antes de os critérios de anulação do trabalho serem atendidos.
- Porcentagem de limite

  Especifica o número total de coisas executadas que iniciam a anulação de um trabalho.

Configuração de tempo limite

Por padrão, não há tempo limite e seu trabalho é executado cancelado ou excluído. Para usar tempos limite, escolhaHabilitar o tempo limiteE, a seguir, especifique um valor de tempo limite entre 1 minuto e 7 dias.

Configuração de repetição

> **Note**
>
> Depois que um trabalho tiver sido criado, o número de novas tentativas não podem ser atualizadas. Você só pode remover a configuração de repetição para todos os tipos de falha. Ao criar um trabalho, considere o número apropriado de novas tentativas a serem usadas para sua configuração. Para evitar custos em excesso devido a possíveis falhas de repetição, adicione uma configuração de cancelamento.

SelecioneAdicionar nova configuraçãoE especifique os seguintes parâmetros para cada configuração:

- Tipo de falha

  Especifica os tipos de falha que devem acionar uma repetição de execução de trabalho. Dentre elas estãoFailed (Com falha),Timeout (Tempo limite), eTudo.
- Número de novas tentativas

  Especifica o número de novas tentativas para o escolhidoTipo de falha. Para ambos os tipos de falha combinados, até 10 novas tentativas podem ser tentadas.

Especifique as configurações de tarefa usando oAWS IoTAPI de trabalhos

Você pode usar oCreateJobou oCreateJobTemplateAPI para especificar as diferentes configurações de trabalho. Veja a seguir como você adiciona essas configurações. Depois de adicionar as configurações, você pode usarJobExecutionSummaryeJobExecutionSummaryForJobpara visualizar o status deles.

Para obter mais informações sobre as diferentes configurações e como isso funciona, consulteComo as configurações de trabalho funcionam (p. 718).

Configuração de implantação

Você pode especificar uma taxa de distribuição constante ou uma taxa de distribuição exponencial para sua configuração de distribuição.

- Definir uma taxa de distribuição constante

Para definir uma taxa de distribuição constante, use o`JobExecutionsRolloutConfig`objeto para adicionar o`maximumPerMinute`Parâmetro para o`CreateJob`solicitação. Esse parâmetro especifica o limite superior da taxa na qual as execuções de trabalho podem ocorrer. Esse valor é opcional e varia de 1 a 1000. Se você não definir o valor, ele usará 1000 como o valor padrão.

```
    "jobExecutionsRolloutConfig": {
        "maximumPerMinute": 1000
    }
```

- Definir uma taxa de distribuição exponencial

  Para definir uma taxa de distribuição de trabalho variável, use o`JobExecutionsRolloutConfig`objeto. Você pode configurar o`ExponentialRolloutRate`propriedade quando você executa o`CreateJob`Operação da API. O exemplo a seguir define uma taxa de implantação exponencial usando a`exponentialRate`parâmetro . Para obter mais informações sobre os parâmetros, consulte`ExponentialRolloutRate`

```
{
...
  "jobExecutionsRolloutConfig": {
    "exponentialRate": {
      "baseRatePerMinute": 50,
      "incrementFactor": 2,
      "rateIncreaseCriteria": {
        "numberOfNotifiedThings": 1000,
        "numberOfSucceededThings": 1000
      },
      "maximumPerMinute": 1000
    }
  }
...
}
```

Onde o parâmetro:

baseRatePerMinute

Especifica a taxa na qual os trabalhos são executados até a`numberOfNotifiedThings`ou`numberOfSucceededThings`o limiar foi atingido.

incrementFactor

Especifica o fator exponencial em que a taxa de implantação aumenta após a`numberOfNotifiedThings`ou`numberOfSucceededThings`o limiar foi atingido.

rateIncreaseCriteria

Especifica o`numberOfNotifiedThings`ou`numberOfSucceededThings`Limite.

Abort configuration

Para adicionar essa configuração usando a API, especifique a`AbortConfig`parâmetro quando você executa o`CreateJob`ou o`CreateJobTemplate`Operação da API. O exemplo a seguir mostra uma configuração de anulação de um trabalho passando por várias execuções com falha, conforme especificado com o`CreateJob`Operação da API.

> **Note**
>
> A exclusão de uma execução de trabalho afeta o valor computacional da execução total concluída. Quando um trabalho é anulado, o serviço cria um `comment` automatizado e um

`reasonCode` para diferenciar um cancelamento orientado pelo usuário de um cancelamento por anulação de trabalho.

```
"abortConfig": {
   "criteriaList": [
      {
         "action": "CANCEL",
         "failureType": "FAILED",
         "minNumberOfExecutedThings": 100,
         "thresholdPercentage": 20
      },
      {
         "action": "CANCEL",
         "failureType": "TIMED_OUT",
         "minNumberOfExecutedThings": 200,
         "thresholdPercentage": 50
      }
   ]
 }
```

Onde o parâmetro:

action

Especifica a ação a ser tomada quando os critérios de anulação são atendidos. Esse parâmetro é necessário, e `CANCEL` é o único valor válido.

failureType

Especifica quais tipos de falha devem iniciar a anulação de um trabalho. Os valores válidos são `FAILED`, `REJECTED`, `TIMED_OUT` e `ALL`.

minNumberOfExecutedThings

Especifica o número de execuções de trabalho concluídas que devem ocorrer antes de os critérios de anulação do trabalho serem atendidos. Neste exemplo, a AWS IoT não verifica se a anulação de um trabalho deve ocorrer até que pelo menos 100 dispositivos tenham concluído as execuções do trabalho.

thresholdPercentage

Especifica o número total de coisas para as quais os trabalhos são executados que podem iniciar a anulação de um trabalho. Neste exemplo,AWS IoTverifica sequencialmente e inicia um cancelamento de trabalho se a porcentagem de limite for atingida. Se pelo menos 20% das execuções completas falharem após 100 execuções serem concluídas, ela cancelará a implantação do trabalho. Se esse critério não for atendido,AWS IoTem seguida, verifica se pelo menos 50% das execuções concluídas expiraram após a conclusão de 200 execuções. Se esse for o caso, ele cancela a implantação do trabalho.

Configuração de tempo limite

Para adicionar essa configuração usando a API, especifique a`TimeoutConfig`parâmetro quando você executa o`CreateJob`ou o`CreateJobTemplate`Operação da API.

Para usar a configuração de tempo limite

1. Para definir o temporizador em andamento ao criar um modelo de trabalho ou trabalho, defina um valor para o`inProgressTimeoutInMinutes`propriedade do opcional`TimeoutConfig`objeto.

```
"timeoutConfig": {
  "inProgressTimeoutInMinutes": number
}
```

2. Para especificar um temporizador de etapa para uma execução de trabalho, defina um valor para`stepTimeoutInMinutes`quando você ligaUpdateJobExecution. O temporizador de etapa se aplica apenas à execução do trabalho que você atualizar. É possível definir um novo valor para esse temporizador cada vez que você atualizar uma execução de trabalho.

   Note

   `UpdateJobExecution` também pode descartar um temporizador de etapa que já foi criado por meio da criação de um novo temporizador de etapa com um valor de -1.

```
{
   ...
    "statusDetails": {
      "string" : "string"
   },
   "stepTimeoutInMinutes": number
}
```

3. Para criar um novo temporizador de etapas, você também pode chamar oStartNextPendingJobExecutionOperação da API.

Configuração de repetição

Note

Ao criar um trabalho, considere o número apropriado de novas tentativas a serem usadas para sua configuração. Para evitar custos em excesso devido a possíveis falhas de repetição, adicione uma configuração de cancelamento. Depois que um trabalho tiver sido criado, o número de novas tentativas não podem ser atualizadas. Você só pode definir o número de novas tentativas para 0 usando aUpdateJobOperação da API.

Para adicionar essa configuração usando a API, especifique a`jobExecutionsRetryConfig`parâmetro quando você executa o`CreateJob`ou o`CreateJobTemplate`Operação da API.

```
{
...
  "jobExecutionsRetryConfig": {
      "criteriaList": [
         {
            "failureType": "string",
            "numberOfRetries": number
         }
      ]
  }
...
}
```

OndecriteriaListé uma matriz que especifica a lista de critérios que determina o número de novas tentativas permitidas para cada tipo de falha para um trabalho.

Dispositivos e trabalhos

Os dispositivos podem se comunicar comAWS IoTTrabalhos usando MQTT, Assinatura HTTP Versão 4 ou HTTP TLS. Para determinar o endpoint a ser usado quando o dispositivo se comunica comAWS IoTJobs, execute oDescribeEndpointPor exemplo, se você executar comando:

```
aws iot describe-endpoint --endpoint-type iot:Data-ATS
```

você obterá um resultado semelhante ao seguinte:

```
{
    "endpointAddress": "a1b2c3d4e5f6g7-ats.iot.us-west-2.amazonaws.com"
}
```

usar o protocolo MQTT

Os dispositivos podem se comunicar comAWS IoTTrabalhos usando o protocolo MQTT. Os dispositivos assinam os tópicos MQTT para serem notificados sobre novos trabalhos e para receberem respostas do serviço Jobs da AWS IoT. Os dispositivos publicam em tópicos MQTT para consultar ou atualizar o estado da execução de um trabalho. Cada dispositivo tem seu próprio tópico geral MQTT. Para obter mais informações sobre publicação e assinatura em tópicos MQTT, consulte Protocolos de comunicação de (p. 83).

Com esse método de comunicação, seu dispositivo usa o certificado de dispositivo específico e a chave privada para autenticar comAWS IoTTrabalhos do.

Seus dispositivos podem se inscrever nos tópicos a seguir.`thing-name`É o nome da coisa associada ao dispositivo.

- **`$aws/things/`*`thing-name`*`/jobs/notify`**

  Subscreva este tópico para notificá-lo quando uma execução de tarefa é adicionada à lista de execuções de trabalho pendentes.
- **`$aws/things/`*`thing-name`*`/jobs/notify-next`**

  Se inscrever neste tópico para notificá-lo quando a próxima execução de trabalho pendente for alterada.
- **`$aws/things/`*`thing-name`*`/`*`request-name`*`/accepted`**

  O serviço Jobs da AWS IoT publica mensagens de êxito e falha em um tópico MQTT. O tópico é formado anexando `accepted` ou `rejected` ao tópico usado para fazer a solicitação. Aqui,`request-name`é o nome de uma solicitação, como`Get`e o tópico pode ser: `$aws/things/myThing/jobs/get`.AWS IoTJobs publica mensagens de sucesso no`$aws/things/myThing/jobs/get/accepted`tópico.
- **`$aws/things/`*`thing-name`*`/`*`request-name`*`/rejected`**

  Aqui,`request-name`é o nome de uma solicitação, como`Get`. Se a solicitação falhar,AWS IoTJobs publica mensagens de falha no`$aws/things/myThing/jobs/get/rejected`tópico.

Você também pode usar as seguintes APIs HTTPS:

- Atualizar o status de uma execução de trabalho chamando a API `UpdateJobExecution`.
- Consultar o status da execução de um trabalho chamando a API `DescribeJobExecution`.
- Recuperar uma lista de execuções de trabalhos pendentes, chamando a API `GetPendingJobExecutions`.
- Recuperar a próxima execução de trabalho pendente chamando o`DescribeJobExecution`API com`jobId`como`$next`.
- Obter e iniciar a próxima execução de trabalho pendente chamando a API `StartNextPendingJobExecution`.

usar HTTP Signature Versão 4

Os dispositivos podem se comunicar comAWS IoTTrabalhos usando HTTP Signature Versão 4 na porta 443. Este é o método usado peloAWSSDKs e CLI. Para obter mais informações sobre essas

ferramentas, consulteAWS CLIReferência de comando: iot-jobs-dataouAWSSDKs e ferramentase consulte o IotJobsDataPlane Seção para seu idioma preferido.

Com esse método de comunicação, seu dispositivo usa as credenciais do IAM para autenticar comAWS IoTTrabalhos do.

Os comandos a seguir estão disponíveis usando esse método:

- DescribeJobExecution

  ```
  aws iot-jobs-data describe-job-execution ...
  ```
- GetPendingJobExecutions

  ```
  aws iot-jobs-data get-pending-job-executions ...
  ```
- StartNextPendingJobExecution

  ```
  aws iot-jobs-data start-next-pending-job-execution ...
  ```
- UpdateJobExecution

  ```
  aws iot-jobs-data update-job-execution ...
  ```

usar HTTP TLS

Os dispositivos podem se comunicar comAWS IoTTrabalhos usando HTTP TLS na porta 8443 usando um cliente de software de terceiros que oferece suporte a esse protocolo.

Com esse método, seu dispositivo usa a autenticação baseada em certificado X.509 (por exemplo, o certificado específico para o dispositivo e a chave privada).

Os comandos a seguir estão disponíveis usando esse método:

- DescribeJobExecution
- GetPendingJobExecutions
- StartNextPendingJobExecution
- UpdateJobExecution

Programar dispositivos para funcionarem com trabalhos

Os exemplos nesta seção usam MQTT para ilustrar como um dispositivo funciona com serviço Jobs da AWS IoT. Como alternativa, você pode usar os comandos da CLI ou a API correspondente. Nesses exemplos, estamos supondo que um dispositivo chamado`MyThing`que assinará os seguintes tópicos MQTT:

- `$aws/things/`*`MyThing`*`/jobs/notify` (ou `$aws/things/`*`MyThing`*`/jobs/notify-next`)
- `$aws/things/`*`MyThing`*`/jobs/get/accepted`
- `$aws/things/`*`MyThing`*`/jobs/get/rejected`
- `$aws/things/`*`MyThing`*`/jobs/`*`jobId`*`/get/accepted`
- `$aws/things/`*`MyThing`*`/jobs/`*`jobId`*`/get/rejected`

Se você estiver usando a assinatura de código paraAWS IoTO código do dispositivo deve verificar a assinatura do seu arquivo de código. A assinatura está no documento de trabalho na propriedade

`codesign`. Para obter mais informações sobre como verificar uma assinatura de arquivo de código, consulte Device Agent Sample.

Tópicos

- Fluxo de trabalho do dispositivo (p. 731)
- Fluxo de trabalho (p. 732)
- Notificações de trabalhos (p. 735)

Fluxo de trabalho do dispositivo

Um dispositivo pode lidar com trabalhos executados usando qualquer uma das seguintes maneiras.

- Obter o próximo trabalho

  1. Quando um dispositivo se torna online pela primeira vez, ele deve assinar o tópico `notify-next` do dispositivo.
  2. Chamar a API MQTT DescribeJobExecution (p. 758) com o jobId `$next` para obter o próximo trabalho, seu documento de trabalho e outros detalhes, incluindo qualquer estado salvo em `statusDetails`. Se o documento de trabalho tiver uma assinatura de arquivo de código, você deverá verificar a assinatura antes de continuar com o processamento da solicitação de trabalho.
  3. Chamar a API MQTT UpdateJobExecution (p. 759) para atualizar o status do trabalho. Ou, para combinar essa e a etapa anterior em uma chamada, o dispositivo pode chamar StartNextPendingJobExecution (p. 757).
  4. (Opcional) Você pode adicionar um temporizador de etapa. Para isso, defina um valor para `stepTimeoutInMinutes` ao chamar UpdateJobExecution (p. 759) ou StartNextPendingJobExecution (p. 757).
  5. Executar as ações especificadas pelo documento de trabalho usando a API MQTT UpdateJobExecution (p. 759) para relatar o andamento do trabalho.
  6. Continue monitorando a execução do trabalho chamando a DescribeJobExecution (p. 758) API MQTT com esse jobId. Se a execução do trabalho for excluída,DescribeJobExecution (p. 758)retorna um`ResourceNotFoundException`.

     O dispositivo deverá conseguir recuperar em um estado válido se a execução do trabalho for cancelada ou excluída enquanto o dispositivo estiver executando a tarefa.
  7. Chame a UpdateJobExecution (p. 759) API MQTT quando finalizar o trabalho para atualizar o respectivo status e relatar êxito ou falha.
  8. Como o status de execução desse trabalho foi alterado para o estado final, o próximo trabalho disponível para execução (se houver) será alterado. O dispositivo é notificado sobre a alteração da próxima execução de trabalho pendente. Neste ponto, o dispositivo deve continuar conforme descrito na etapa 2.

  Se o dispositivo permanecer online, ele continuará a receber notificações da próxima execução de trabalho pendente, incluindo os dados de execução do trabalho, quando concluir um trabalho ou uma nova execução de trabalho pendente for adicionada. Quando isso ocorre, o dispositivo continua, conforme descrito na etapa 2.

- Escolha entre os trabalhos disponíveis

  1. Quando um dispositivo se torna online pela primeira vez, ele deve assinar o tópico `notify` da coisa.
  2. Chamar a API MQTT GetPendingJobExecutions (p. 757) para obter uma lista de execuções de trabalhos pendentes.
  3. Se a lista contiver uma ou mais execuções de trabalho, selecionar uma.

4. Chamar a API MQTT DescribeJobExecution (p. 758) para obter o documento de trabalho e outros detalhes, incluindo qualquer estado salvo em `statusDetails`.
5. Chamar a API MQTT UpdateJobExecution (p. 759) para atualizar o status do trabalho. Se o campo `includeJobDocument` estiver definido como `true` nesse comando, o dispositivo poderá ignorar a etapa anterior e recuperar o documento de trabalho neste ponto.
6. Você pode adicionar um temporizador de etapa. Para isso, defina um valor para `stepTimeoutInMinutes` ao chamar UpdateJobExecution (p. 759).
7. Executar as ações especificadas pelo documento de trabalho usando a API MQTT UpdateJobExecution (p. 759) para relatar o andamento do trabalho.
8. Continue monitorando a execução do trabalho chamando a DescribeJobExecution (p. 758) API MQTT com esse jobId. Se a execução do trabalho for cancelada ou excluída enquanto o dispositivo estiver executando o trabalho, o dispositivo deve conseguir voltar para um estado válido.
9. Chame a UpdateJobExecution (p. 759) API MQTT quando finalizar o trabalho para atualizar o respectivo status e relatar êxito ou falha.

Se o dispositivo permanecer online, ele será notificado de todas as execuções de trabalho pendentes quando uma nova execução de trabalho se tornar disponível. Quando isso ocorrer, o dispositivo poderá continuar, conforme descrito na etapa 2.

Se o dispositivo não puder executar o trabalho, ele deverá chamar a API MQTT UpdateJobExecution (p. 759) para atualizar o status do trabalho para `REJECTED`.

Fluxo de trabalho

A seguir, mostra as diferentes etapas no fluxo de trabalho de trabalhos, desde o início de um novo trabalho até o relatório do status de conclusão de uma execução de trabalho.

Iniciar um novo trabalho

Quando um novo trabalho é criado,AWS IoTJobs publicará uma mensagem no`$aws/things/`*`thing-name`*`/jobs/notify`tópico para cada dispositivo de destino.

A mensagem contém as seguintes informações:

```
{
    "timestamp":1476214217017,
    "jobs":{
        "QUEUED":[{
            "jobId":"0001",
            "queuedAt":1476214216981,
            "lastUpdatedAt":1476214216981,
            "versionNumber" : 1
        }]
    }
}
```

O dispositivo recebe essa mensagem no tópico `'$aws/things/`*`thingName`*`/jobs/notify'` quando a execução do trabalho é enfileirada.

Obter informações do trabalho

Para obter mais informações sobre a execução de um trabalho, o dispositivo chama a API MQTT DescribeJobExecution (p. 758) com o campo `includeJobDocument` definido como `true` (o padrão).

Se a solicitação for bem-sucedida, o serviço Jobs da AWS IoT publicará uma mensagem no tópico `$aws/things/MyThing/jobs/0023/get/accepted`:

```
{
    "clientToken" : "client-001",
    "timestamp" : 1489097434407,
    "execution" : {
        "approximateSecondsBeforeTimedOut": number,
        "jobId" : "023",
        "status" : "QUEUED",
        "queuedAt" : 1489097374841,
        "lastUpdatedAt" : 1489097374841,
        "versionNumber" : 1,
        "jobDocument" : {
            < contents of job document >
        }
    }
}
```

Se a solicitação falhar, o serviço Jobs da AWS IoT publicará uma mensagem no tópico `$aws/things/MyThing/jobs/0023/get/rejected`.

O dispositivo agora tem o documento de trabalho, que ele pode usar para executar as operações remotas do trabalho. Se o documento de trabalho contiver uma pré-signed URL do Amazon S3, o dispositivo poderá usar essa URL para fazer download de todos os arquivos necessários do trabalho.

Relatar o status da execução de um trabalho

Enquanto executa o trabalho, o dispositivo pode chamar a API MQTT UpdateJobExecution (p. 759) para atualizar o status da execução do trabalho.

Por exemplo, um dispositivo pode atualizar o status da execução do trabalho para `IN_PROGRESS` publicando a seguinte mensagem no tópico `$aws/things/MyThing/jobs/0023/update`:

```
{
    "status":"IN_PROGRESS",
    "statusDetails": {
        "progress":"50%"
    },
    "expectedVersion":"1",
    "clientToken":"client001"
}
```

O serviço Jobs responde publicando uma mensagem no tópico `$aws/things/MyThing/jobs/0023/update/accepted` ou `$aws/things/MyThing/jobs/0023/update/rejected`:

```
{
    "clientToken":"client001",
    "timestamp":1476289222841
}
```

O dispositivo pode combinar as duas solicitações anteriores chamando StartNextPendingJobExecution (p. 757). Isso obtém e inicia a próxima execução de trabalho pendente e permite que o dispositivo atualize o status de execução do trabalho. Essa solicitação também retorna o documento de trabalho quando há uma execução de trabalho pendente.

Se o trabalho contém um TimeoutConfig, o temporizador em andamento começará a ser executado. Você também pode definir um temporizador de etapa para uma execução de trabalho. Para isso, defina um valor para `stepTimeoutInMinutes` ao chamar UpdateJobExecution. O temporizador de etapa se aplica apenas à execução do trabalho que você atualizar. É possível definir um novo valor para esse

temporizador cada vez que você atualizar uma execução de trabalho. Você também pode criar um temporizador de etapa ao chamar StartNextPendingJobExecution. Se a execução de trabalho permanecer com o status `IN_PROGRESS` por mais tempo que o intervalo do temporizador de etapa, ela falhará e alternará para o status `TIMED_OUT` terminal. O temporizador de etapa não tem efeito sobre o temporizador em andamento que você define ao criar um trabalho.

O campo `status` pode ser definido como `IN_PROGRESS`, `SUCCEEDED` ou `FAILED`. Você não pode atualizar o status da execução de um trabalho que já esteja em um estado terminal.

Execução do relatório concluída

Quando o dispositivo conclui a execução do trabalho, ele chama a API MQTT UpdateJobExecution (p. 759). Se o trabalho tiver sido bem-sucedido, defina `status` como `SUCCEEDED` e, na carga da mensagem, em `statusDetails`, adicione outras informações sobre o trabalho, como pares nome/valor. Os temporizadores de etapa e em andamento terminam quando a execução do trabalho é concluída.

Por exemplo:

```
{
    "status":"SUCCEEDED",
    "statusDetails": {
        "progress":"100%"
    },
    "expectedVersion":"2",
    "clientToken":"client-001"
}
```

Se o trabalho não foi bem-sucedido, defina `status` como `FAILED` e, em `statusDetails`, adicione informações sobre o erro que ocorreu:

```
{
    "status":"FAILED",
    "statusDetails": {
        "errorCode":"101",
        "errorMsg":"Unable to install update"
    },
    "expectedVersion":"2",
    "clientToken":"client-001"
}
```

> **Note**
>
> O atributo `statusDetails` pode conter qualquer número de pares de nome e valor.

Quando o serviço Jobs da AWS IoT recebe essa atualização, ele publica uma mensagem no tópico `$aws/things/MyThing/jobs/notify` para indicar que a execução do trabalho foi concluída:

```
{
    "timestamp":1476290692776,
    "jobs":{}
}
```

Trabalhos adicionais

Se houver outras execuções de trabalho pendentes para o dispositivo, elas serão incluídas na mensagem publicada em `$aws/things/MyThing/jobs/notify`.

Por exemplo:

```
{
    "timestamp":1476290692776,
    "jobs":{
        "QUEUED":[{
            "jobId":"0002",
            "queuedAt":1476290646230,
            "lastUpdatedAt":1476290646230
        }],
        "IN_PROGRESS":[{
            "jobId":"0003",
            "queuedAt":1476290646230,
            "lastUpdatedAt":1476290646230
        }]
    }
}
```

Notificações de trabalhos

O serviço Jobs da AWS IoT publica mensagens MQTT para tópicos reservadas quando os trabalhos estão pendentes ou quando a execução do primeiro trabalho da lista é alterada. Os dispositivos podem acompanhar trabalhos pendentes assinando esses tópicos.

Tipos de notificação de Job

As notificações de trabalhos são publicadas em tópicos MQTT como cargas JSON. Há dois tipos de notificações:

- A `ListNotification` contém uma lista de até 10 execuções de trabalhos pendentes. As execuções de trabalhos nessa lista têm valores de status de `IN_PROGRESS` ou `QUEUED`. Elas são classificados por status (execuções de trabalho de `IN_PROGRESS` antes de execuções de trabalhos de `QUEUED`) e, em seguida, pelo número de vezes em que foram colocadas em fila.

  Uma `ListNotification` é publicada sempre que um dos seguintes critérios são atendidos.
  - Uma nova execução de trabalho é colocada na fila ou muda para um status terminal (`IN_PROGRESS` ou `QUEUED`).
  - Uma execução de trabalho anterior muda para um status final (`FAILED`, `SUCCEEDED`, `CANCELED`, `TIMED_OUT`, `REJECTED` ou `REMOVED`).
- A `NextNotification` contém informações resumidas sobre a próxima execução de trabalho da fila.

  Uma `NextNotification` é publicada sempre que a primeira execução de trabalho da lista é alterada.
  - Uma nova execução é adicionada à lista como `QUEUED` e é a primeira da lista.
  - O status de uma execução de trabalho existente que não é a primeira da lista é alterado de `QUEUED` para `IN_PROGRESS` e torna-se a primeira na lista. (Isso acontece quando não há outras execuções de trabalho de `IN_PROGRESS` na lista ou quando a execução de trabalho cujo status é alterado de `QUEUED` para `IN_PROGRESS` foi colocado na fila antes de qualquer outra execução de `IN_PROGRESS` na lista.)
  - O status da primeira execução de trabalho da lista é alterado para um status terminal e é removido da lista.

Para obter mais informações sobre publicação e assinatura em tópicos MQTT, consulte the section called “Protocolos de comunicação de” (p. 83).

> **Note**
>
> As notificações não estão disponíveis quando você usa HTTP Signature Versão 4 ou HTTP TLS para se comunicar com os trabalhos.

Job pendente

O serviço Jobs da AWS IoT publica uma mensagem em um tópico MQTT quando um trabalho é adicionado ou removido da lista de execuções de trabalhos pendentes de coisa, ou quando há uma alteração na ordem dos trabalhos na lista:

- `$aws/things/thingName/jobs/notify`
- `$aws/things/thingName/jobs/notify-next`

As mensagens contêm as seguintes cargas de exemplo:

`$aws/things/thingName/jobs/notify`:

```
{
  "timestamp" : 10011,
  "jobs" : {
    "IN_PROGRESS" : [ {
      "jobId" : "other-job",
      "queuedAt" : 10003,
      "lastUpdatedAt" : 10009,
      "executionNumber" : 1,
      "versionNumber" : 1
    } ],
    "QUEUED" : [ {
      "jobId" : "this-job",
      "queuedAt" : 10011,
      "lastUpdatedAt" : 10011,
      "executionNumber" : 1,
      "versionNumber" : 0
    } ]
  }
}
```

`$aws/things/thingName/jobs/notify-next`:

```
{
  "timestamp" : 10011,
  "execution" : {
    "jobId" : "other-job",
    "status" : "IN_PROGRESS",
    "queuedAt" : 10009,
    "lastUpdatedAt" : 10009,
    "versionNumber" : 1,
    "executionNumber" : 1,
    "jobDocument" : {"c":"d"}
  }
}
```

Os possíveis valores do status da execução do trabalho são `QUEUED`, `IN_PROGRESS`, `FAILED`, `SUCCEEDED`, `CANCELED`, `TIMED_OUT`, `REJECTED` e `REMOVED`.

A série de exemplos a seguir mostra as notificações publicadas em cada tópico quando as execuções de um trabalho são criadas e mudam de um estado para outro.

Primeiro, é criado um trabalho chamado `job1`. Essa notificação é publicada no tópico `jobs/notify`:

```
{
  "timestamp": 1517016948,
  "jobs": {
```

```
    "QUEUED": [
      {
        "jobId": "job1",
        "queuedAt": 1517016947,
        "lastUpdatedAt": 1517016947,
        "executionNumber": 1,
        "versionNumber": 1
      }
    ]
  }
}
```

Essa notificação é publicada no tópico `jobs/notify-next`:

```
{
  "timestamp": 1517016948,
  "execution": {
    "jobId": "job1",
    "status": "QUEUED",
    "queuedAt": 1517016947,
    "lastUpdatedAt": 1517016947,
    "versionNumber": 1,
    "executionNumber": 1,
    "jobDocument": {
      "operation": "test"
    }
  }
}
```

Ao criar outro trabalho (`job2`), essa notificação é publicada no tópico `jobs/notify`:

```
{
  "timestamp": 1517017192,
  "jobs": {
    "QUEUED": [
      {
        "jobId": "job1",
        "queuedAt": 1517016947,
        "lastUpdatedAt": 1517016947,
        "executionNumber": 1,
        "versionNumber": 1
      },
      {
        "jobId": "job2",
        "queuedAt": 1517017191,
        "lastUpdatedAt": 1517017191,
        "executionNumber": 1,
        "versionNumber": 1
      }
    ]
  }
}
```

Não é publicada uma notificação no tópico `jobs/notify-next` porque o próximo trabalho da fila (`job1`) não sofreu alteração. Quando `job1` começa a executar, o status muda para `IN_PROGRESS`. Não são publicadas notificações porque a lista de trabalhos e o próximo trabalho na fila não sofreram alterações.

Ao adicionar um terceiro trabalho (`job3`), essa notificação é publicada no tópico `jobs/notify`:

```
{
  "timestamp": 1517017906,
```

```
  "jobs": {
    "IN_PROGRESS": [
      {
        "jobId": "job1",
        "queuedAt": 1517016947,
        "lastUpdatedAt": 1517017472,
        "startedAt": 1517017472,
        "executionNumber": 1,
        "versionNumber": 2
      }
    ],
    "QUEUED": [
      {
        "jobId": "job2",
        "queuedAt": 1517017191,
        "lastUpdatedAt": 1517017191,
        "executionNumber": 1,
        "versionNumber": 1
      },
      {
        "jobId": "job3",
        "queuedAt": 1517017905,
        "lastUpdatedAt": 1517017905,
        "executionNumber": 1,
        "versionNumber": 1
      }
    ]
  }
}
```

Não é publicada uma notificação no tópico `jobs/notify-next` porque o próximo trabalho da fila ainda é `job1`.

Quando `job1` for concluído, o status mudará para `SUCCEEDED` e essa notificação será publicada no tópico `jobs/notify`:

```
{
  "timestamp": 1517186269,
  "jobs": {
    "QUEUED": [
      {
        "jobId": "job2",
        "queuedAt": 1517017191,
        "lastUpdatedAt": 1517017191,
        "executionNumber": 1,
        "versionNumber": 1
      },
      {
        "jobId": "job3",
        "queuedAt": 1517017905,
        "lastUpdatedAt": 1517017905,
        "executionNumber": 1,
        "versionNumber": 1
      }
    ]
  }
}
```

A essa altura, `job1` já terá sido removido da fila e o próximo trabalho a ser executado será `job2`. Essa notificação é publicada no tópico `jobs/notify-next`:

```
{
  "timestamp": 1517186269,
```

```
  "execution": {
    "jobId": "job2",
    "status": "QUEUED",
    "queuedAt": 1517017191,
    "lastUpdatedAt": 1517017191,
    "versionNumber": 1,
    "executionNumber": 1,
    "jobDocument": {
      "operation": "test"
    }
  }
}
```

Se o `job3` precisar começar antes do `job2` (o que não é recomendado), o status do `job3` será alterado para `IN_PROGRESS`. Se isso acontecer, o `job2` não será mais o próximo na fila, e essa notificação será publicada no tópico `jobs/notify-next`:

```
{
  "timestamp": 1517186779,
  "execution": {
    "jobId": "job3",
    "status": "IN_PROGRESS",
    "queuedAt": 1517017905,
    "startedAt": 1517186779,
    "lastUpdatedAt": 1517186779,
    "versionNumber": 2,
    "executionNumber": 1,
    "jobDocument": {
      "operation": "test"
    }
  }
}
```

Nenhuma notificação é publicada no tópico `jobs/notify` porque nenhum trabalho foi adicionado ou removido.

Se o dispositivo rejeitar o `job2` e atualizar seu status para `REJECTED`, essa notificação será publicada no tópico `jobs/notify`:

```
{
  "timestamp": 1517189392,
  "jobs": {
    "IN_PROGRESS": [
      {
        "jobId": "job3",
        "queuedAt": 1517017905,
        "lastUpdatedAt": 1517186779,
        "startedAt": 1517186779,
        "executionNumber": 1,
        "versionNumber": 2
      }
    ]
  }
}
```

Se a exclusão do `job3` (que ainda está em andamento) for forçada, essa notificação será publicada no tópico `jobs/notify`:

```
{
  "timestamp": 1517189551,
  "jobs": {}
```

```
}
```

Nesse ponto, a fila está vazia. Essa notificação é publicada no tópico `jobs/notify-next`:

```
{
  "timestamp": 1517189551
}
```

AWS IoTAPIs de trabalhos

AWS IoTA API Jobs pode ser usada para qualquer uma das seguintes categorias:

- As operações da API usadas para tarefas administrativas, como gerenciamento e controle de trabalhos. Este é oplano de controle.
- Operações de API usadas para dispositivos que executam esses trabalhos. Este é oplano de dados, que permite enviar e receber dados.

O gerenciamento e o controle de Job usam uma API de protocolo HTTPS. Os dispositivos podem usar uma API MQTT ou de protocolo HTTPS. A API do plano de controle é projetada para um volume baixo de chamadas típicas durante a criação e o acompanhamento de trabalhos. Normalmente, ela abre uma conexão para uma única solicitação e, em seguida, fecha a conexão depois que a resposta é recebida. O plano de dados HTTPS e a API MQTT permitem sondagem longa. Essas operações da API são projetadas para grandes quantidades de tráfego que podem ser escaladas para milhões de dispositivos.

Cada API HTTPS do serviço Jobs da AWS IoT tem um comando correspondente que permite chamar a API na AWS CLI Os comandos são em letras minúsculas, com hifens entre as palavras que compõem o nome da API. Por exemplo, você pode chamar a API `CreateJob` na CLI, digitando:

```
aws iot create-job ...
```

Se um erro ocorrer durante uma operação, você receberá uma resposta de erro que contém as informações sobre o erro.

ErrorResponse

Contém informações sobre um erro que ocorreu durante uma operação do serviço Jobs da AWS IoT.

O exemplo a seguir mostra a sintaxe dessa operação:

```
{
    "code": "ErrorCode",
    "message": "string",
    "clientToken": "string",
    "timestamp": timestamp,
    "executionState": JobExecutionState
}
```

Veja a seguir uma descrição deste`ErrorResponse`:

`code`

O ErrorCode pode ser definido como:

InvalidTopic

A solicitação foi enviada a um tópico no namespace do serviço Jobs da AWS IoT que não está mapeada para nenhuma API.

InvalidJson

O conteúdo da solicitação não pôde ser interpretado como JSON codificado em UTF-8 válido.

InvalidRequest

O conteúdo da solicitação era inválido. Por exemplo, esse código é retornado quando uma solicitação `UpdateJobExecution` contém detalhes do status inválido. A mensagem contém detalhes sobre o erro.

InvalidStateTransition

Uma atualização tentou alterar a execução do trabalho para um estado que é inválido por causa do estado atual da execução do trabalho (por exemplo, uma tentativa de alterar uma solicitação no estado SUCCEEDED para o estado IN_PROGRESS). Nesse caso, o corpo da mensagem de erro também contém o campo `executionState`.

ResourceNotFound

A `JobExecution` especificada pelo tópico da solicitação não existe.

VersionMismatch

A versão esperada especificada na solicitação não corresponde à versão da execução do trabalho no serviço Jobs da AWS IoT. Nesse caso, o corpo da mensagem de erro também contém o campo `executionState`.

InternalError

Ocorreu um erro interno durante o processamento da solicitação.

RequestThrottled

A solicitação foi acelerada.

TerminalStateReached

Ocorre quando um comando para descrever um trabalho é executado em um trabalho que está em um estado terminal.

`message`

A sequência de uma mensagem de erro.

`clientToken`

Uma sequência arbitrária usada para correlacionar uma solicitação com sua resposta.

`timestamp`

O tempo, em segundos, desde a epoch.

`executionState`

Um objeto `JobExecutionState`. Esse campo é incluído apenas quando o campo `code` tem o valor `InvalidStateTransition` ou `VersionMismatch`. Nesses casos, torna-se desnecessário executar uma solicitação `DescribeJobExecution` separada para obter os dados do status da execução do trabalho atual.

A seguir lista as APIs de trabalhos e os tipos de dados.

- API e tipos de dados de gerenciamento e controle de trabalhos (p. 742)

- APIs e tipos de dados MQTT e HTTPS de dispositivos de trabalho (p. 755)

API e tipos de dados de gerenciamento e controle de trabalhos

Os comandos a seguir estão disponíveis para gerenciamento e controle de Job na CLI e por meio do protocolo HTTPS.

- Tipos de dados de gerenciamento e controle de trabalhos (p. 742)
- Operações de API de gerenciamento e controle de Job (p. 745)

Para determinar o*`endpoint-url`*parâmetro para seus comandos da CLI, execute este comando.

```
aws iot describe-endpoint --endpoint-type=iot:Jobs
```

Este comando retorna a seguinte saída.

```
{
"endpointAddress": "account-specific-prefix.jobs.iot.aws-region.amazonaws.com"
}
```

> **Note**
>
> O endpoint Jobs não oferece suporte a ALPN`z-amzn-http-ca`.

Tipos de dados de gerenciamento e controle de trabalhos

Os seguintes tipos de dados são usados pelos aplicativos de gerenciamento e controle para comunicação com oAWS IoTTrabalhos.

Trabalho

O objeto `Job` contém detalhes sobre um trabalho. Veja a seguir a sintaxe:

```
{
    "jobArn": "string",
    "jobId": "string",
    "status": "IN_PROGRESS|CANCELED|SUCCEEDED",
    "forceCanceled": boolean,
    "targetSelection": "CONTINUOUS|SNAPSHOT",
    "comment": "string",
    "targets": ["string"],
    "description": "string",
    "createdAt": timestamp,
    "lastUpdatedAt": timestamp,
    "completedAt": timestamp,
    "jobProcessDetails": {
        "processingTargets": ["string"],
        "numberOfCanceledThings": long,
        "numberOfSucceededThings": long,
        "numberOfFailedThings": long,
        "numberOfRejectedThings": long,
        "numberOfQueuedThings": long,
        "numberOfInProgressThings": long,
        "numberOfRemovedThings": long,
```

```
        "numberOfTimedOutThings": long
    },
    "presignedUrlConfig": {
        "expiresInSec": number,
        "roleArn": "string"
    },
    "jobExecutionsRolloutConfig": {
        "exponentialRate": {
           "baseRatePerMinute": integer,
           "incrementFactor": integer,
           "rateIncreaseCriteria": {
              "numberOfNotifiedThings": integer, // Set one or the other
              "numberOfSucceededThings": integer // of these two values.
           },
           "maximumPerMinute": integer
      }
    },
    "abortConfig": {
       "criteriaList": [
          {
             "action": "string",
             "failureType": "string",
             "minNumberOfExecutedThings": integer,
             "thresholdPercentage": integer
          }
       ]
    },
    "timeoutConfig": {
        "inProgressTimeoutInMinutes": long
    }
}
```

Para obter mais informações, consulte`Job`ou`job`

JobSummary

O objeto `JobSummary` contém um resumo do trabalho. Veja a seguir a sintaxe:

```
{
    "jobArn": "string",
    "jobId": "string",
    "status": "IN_PROGRESS|CANCELED|SUCCEEDED",
    "targetSelection": "CONTINUOUS|SNAPSHOT",
    "thingGroupId": "string",
    "createdAt": timestamp,
    "lastUpdatedAt": timestamp,
    "completedAt": timestamp
}
```

Para obter mais informações, consulte`JobSummary`ou`job-summary`

JobExecution

O objeto `JobExecution` representa a execução de um trabalho em um dispositivo. Veja a seguir a sintaxe:

```
{
    "approximateSecondsBeforeTimedOut": 50,
    "executionNumber": 1234567890,
    "forceCanceled": true|false,
    "jobId": "string",
    "lastUpdatedAt": timestamp,
    "queuedAt": timestamp,
```

```
    "startedAt": timestamp,
    "status": "QUEUED|IN_PROGRESS|FAILED|SUCCEEDED|CANCELED|TIMED_OUT|REJECTED|REMOVED",
    "forceCanceled": boolean,
    "statusDetails": {
        "detailsMap": {
            "string": "string" ...
        },
        "status": "string"
    },
    "thingArn": "string",
    "versionNumber": 123
}
```

Para obter mais informações, consulte `JobExecution` ou `job-execution`.

JobExecutionSummary

O`JobExecutionSummary`O objeto contém informações resumidas sobre a execução do trabalho. Veja a seguir a sintaxe:

```
{
    "executionNumber": 1234567890,
    "queuedAt": timestamp,
    "lastUpdatedAt": timestamp,
    "startedAt": timestamp,
    "status": "QUEUED|IN_PROGRESS|FAILED|SUCCEEDED|CANCELED|TIMED_OUT|REJECTED|REMOVED"
}
```

Para obter mais informações, consulte `JobExecutionSummary` ou `job-execution-summary`.

JobExecutionSummaryForJob

O objeto `JobExecutionSummaryForJob` contém um resumo das informações sobre execuções de trabalho de um trabalho específico. Veja a seguir a sintaxe:

```
{
    "executionSummaries": [
        {
            "thingArn": "arn:aws:iot:us-west-2:123456789012:thing/MyThing",
            "jobExecutionSummary": {
                "status": "IN_PROGRESS",
                "lastUpdatedAt": 1549395301.389,
                "queuedAt": 1541526002.609,
                "executionNumber": 1
            }
        },
        ...
    ]
}
```

Para obter mais informações, consulte `JobExecutionSummaryForJob` ou `job-execution-summary-for-job`.

JobExecutionSummaryForThing

O objeto `JobExecutionSummaryForThing` contém um resumo das informações sobre a execução de um trabalho de uma coisa específica. Veja a seguir a sintaxe:

```
{
    "executionSummaries": [
        {
```

```
            "jobExecutionSummary": {
                "status": "IN_PROGRESS",
                "lastUpdatedAt": 1549395301.389,
                "queuedAt": 1541526002.609,
                "executionNumber": 1
            },
            "jobId": "MyThingJob"
        },
        ...
    ]
}
```

Para obter mais informações, consulte `JobExecutionSummaryForThing` ou `job-execution-summary-for-thing`.

Operações de API de gerenciamento e controle de Job

Use as seguintes operações de API ou comandos da CLI:

AssociateTargetsWithJob

Associa um grupo a um trabalho contínuo. Os seguintes critérios devem ser atendidos:

- O trabalho deve ter sido criado com o campo `targetSelection` definido como `CONTINUOUS`.
- O status do trabalho deve ser `IN_PROGRESS`.
- O número total de destinos associados a um trabalho não deve ultrapassar 100.

HTTPS request

```
POST /jobs/jobId/targets

{
"targets": [ "string" ],
"comment": "string"
}
```

Para obter mais informações, consulte`AssociateTargetsWithJob`

CLI syntax

```
aws iot  associate-targets-with-job \
--targets <value> \
--job-id <value> \
[--comment <value>]  \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

`cli-input-json` format:

```
{
"targets": [
"string"
],
"jobId": "string",
"comment": "string"
}
```

Para obter mais informações, consulte`associate-targets-with-job`

CancelJob

Cancela um trabalho.

HTTPS request

```
PUT /jobs/jobId/cancel

{
"force": boolean,
"comment": "string",
"reasonCode": "string"
}
```

Para obter mais informações, consulte`CancelJob`

CLI syntax

```
aws iot cancel-job \
    --job-id <value> \
    [--force <value>]  \
    [--comment <value>]  \
    [--reasonCode <value>]  \
    [--cli-input-json <value>] \
    [--generate-cli-skeleton]
```

`cli-input-json` format:

```
{
    "jobId": "string",
    "force": boolean,
    "comment": "string"
}
```

Para obter mais informações, consulte`cancel-job`

CancelJobExecution

Cancela uma execução de trabalho em um dispositivo.

HTTPS request

```
PUT /things/thingName/jobs/jobId/cancel

{
"force": boolean,
"expectedVersion": "string",
"statusDetails": {
    "string": "string"
    ...
}
}
```

Para obter mais informações, consulte`CancelJobExecution`

CLI syntax

```
aws iot cancel-job-execution \
--job-id <value> \
```

```
--thing-name <value> \
[--force | --no-force] \
[--expected-version <value>] \
[--status-details <value>]  \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

`cli-input-json` format:

```
{
"jobId": "string",
"thingName": "string",
"force": boolean,
"expectedVersion": long,
"statusDetails": {
"string": "string"
}
}
```

Para obter mais informações, consulte`cancel-job-execution`

CreateJob

Cria um trabalho. Você pode fornecer o documento de trabalho como um link para um arquivo em um bucket do Amazon S3 (`documentSource`parâmetro) ou no corpo da solicitação (`document`parâmetro).

Um trabalho pode se tornar contínuo definindo o parâmetro `targetSelection` opcional como `CONTINUOUS`. (O padrão é `SNAPSHOT`.) Um trabalho contínuo pode ser usado para integrar ou atualizar dispositivos quando são adicionados a um grupo, porque ele continua a ser executado em coisas recém-adicionadas, mesmo depois que as coisas no grupo no momento em que o trabalho foi criado tiverem concluído o trabalho.

Um trabalho pode ter um TimeoutConfig opcional, que define o valor do temporizador em andamento. O temporizador em andamento não pode ser atualizado e é aplicado a todas as execuções do trabalho.

As seguintes validações são realizadas em argumentos para a API `CreateJob`:

- O argumento `targets` deve ser uma lista de ARNs válidos de coisas ou de grupos de coisas. Todas as coisas e grupos de coisas devem estar no seuConta da AWS.
- O`documentSource`O argumento deve ser uma URL válida do Amazon S3 para um documento de trabalho. As URLs do Amazon S3 têm o formato:`https://s3.amazonaws.com/`*`bucketName`*`/`*`objectName`*.
- O documento armazenado na URL especificada pelo argumento `documentSource` deve ser um documento JSON codificado em UTF-8.
- O tamanho de um documento de trabalho é limitado a 32 KB devido ao limite do tamanho de uma mensagem MQTT (128 KB) e da criptografia.
- O`jobId`Deve ser exclusivo em seuConta da AWS.

HTTPS request

```
PUT /jobs/jobId

{
"targets": [ "string" ],
"document": "string",
"documentSource": "string",
"description": "string",
```

```
"jobTemplateArn": "string",
"presignedUrlConfigData": {
    "roleArn": "string",
    "expiresInSec": "integer"
},
"targetSelection": "CONTINUOUS|SNAPSHOT",
"jobExecutionsRolloutConfig": {
    "exponentialRate": {
       "baseRatePerMinute": integer,
       "incrementFactor": integer,
       "rateIncreaseCriteria": {
          "numberOfNotifiedThings": integer, // Set one or the other
          "numberOfSucceededThings": integer // of these two values.
       },
       "maximumPerMinute": integer
  }
},
"abortConfig": {
   "criteriaList": [
      {
         "action": "string",
         "failureType": "string",
         "minNumberOfExecutedThings": integer,
         "thresholdPercentage": integer
      }
   ]
},
"timeoutConfig": {
  "inProgressTimeoutInMinutes": long
}
}
```

Para obter mais informações, consulte`CreateJob`

CLI syntax

```
aws iot create-job \
    --job-id <value> \
    --targets <value> \
    [--document-source <value>] \
    [--document <value>] \
    [--description <value>] \
    [--job-template-arn <value>] \
    [--presigned-url-config <value>] \
    [--target-selection <value>] \
    [--job-executions-rollout-config <value>] \
    [--abort-config <value>] \
    [--timeout-config <value>] \
    [--document-parameters <value>]  \
    [--cli-input-json <value>] \
    [--generate-cli-skeleton]
```

`cli-input-json` format:

```
{
    "jobId": "string",
    "targets": [ "string" ],
    "documentSource": "string",
    "document": "string",
    "description": "string",
    "jobTemplateArn": "string",
    "presignedUrlConfig": {
        "roleArn": "string",
        "expiresInSec": long
```

```
     },
    "targetSelection": "string",
    "jobExecutionsRolloutConfig": {
          "exponentialRate": {
              "baseRatePerMinute": integer,
              "incrementFactor": integer,
              "rateIncreaseCriteria": {
                 "numberOfNotifiedThings": integer, // Set one or the other
                 "numberOfSucceededThings": integer // of these two values.
              },
      "maximumPerMinute": integer
      }
    },
    "abortConfig": {
    "criteriaList": [
        {
           "action": "string",
           "failureType": "string",
           "minNumberOfExecutedThings": integer,
           "thresholdPercentage": integer
         }
      ]
    },
    "timeoutConfig": {
          "inProgressTimeoutInMinutes": long
    },
    "documentParameters": {
    "string": "string"
    }
}
```

Para obter mais informações, consulte`create-job`

DeleteJob

Exclui um trabalho e as execuções de trabalho correspondentes.

A exclusão de um trabalho pode levar tempo, dependendo do número de execuções criadas para o trabalho e de vários outros fatores. Enquanto o trabalho está sendo excluído, seu status é mostrado como "DELETION_IN_PROGRESS". A tentativa de excluir ou cancelar um trabalho cujo status já seja "DELETION_IN_PROGRESS" resulta em um erro.

HTTPS request

```
DELETE /jobs/jobId?force=force
```

Para obter mais informações, consulte`DeleteJob`

CLI syntax

```
aws iot  delete-job \
--job-id <value> \
[--force | --no-force]  \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

`cli-input-json` format:

```
{
"jobId": "string",
```

```
"force": boolean
}
```

Para obter mais informações, consulte`delete-job`

DeleteJobExecution

Exclui uma execução de trabalho.

HTTPS request

```
DELETE /things/thingName/jobs/jobId/executionNumber/executionNumber?force=force
```

Para obter mais informações, consulte`DeleteJobExecution`

CLI syntax

```
aws iot  delete-job-execution \
--job-id <value> \
--thing-name <value> \
--execution-number <value> \
[--force | --no-force]  \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

`cli-input-json` format:

```
{
"jobId": "string",
"thingName": "string",
"executionNumber": long,
"force": boolean
}
```

Para obter mais informações, consulte`delete-job-execution`

DescribeJob

Obtém os detalhes da execução do trabalho.

HTTPS request

```
GET /jobs/jobId
```

Para obter mais informações, consulte`DescribeJob`

CLI syntax

```
aws iot describe-job \
--job-id <value>  \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

`cli-input-json` format:

```
{
```

```
"jobId": "string"
}
```

Para obter mais informações, consulte`describe-job`

DescribeJobExecution

Obtém os detalhes da execução de um trabalho. O status da execução do trabalho deve ser `SUCCEEDED` ou `FAILED`.

HTTPS request

```
GET /things/thingName/jobs/jobId?executionNumber=executionNumber
```

Para obter mais informações, consulte`DescribeJobExecution`

CLI syntax

```
aws iot  describe-job-execution \
--job-id <value> \
--thing-name <value> \
[--execution-number <value>]  \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

`cli-input-json` format:

```
{
"jobId": "string",
"thingName": "string",
"executionNumber": long
}
```

Para obter mais informações, consulte`describe-job-execution`

GetJobDocument

Obtém o documento de trabalho para um trabalho.

> **Note**
>
> URLs de espaço reservado não são substituídas por URLs pré-assinadas do Amazon S3 no documento retornado. As pre-signed URLs são geradas apenas quando o serviço Jobs da AWS IoT recebe uma solicitação por meio do MQTT.

HTTPS request

```
GET /jobs/jobId/job-document
```

Para obter mais informações, consulte`GetJobDocument`

CLI syntax

```
aws iot get-job-document \
--job-id <value>  \
[--cli-input-json <value>] \
```

```
[--generate-cli-skeleton]
```

`cli-input-json` format:

```
{
"jobId": "string"
}
```

Para obter mais informações, consulte`get-job-document`

ListJobExecutionsForJob

Obtém uma lista de execuções de trabalhos para um trabalho.

HTTPS request

```
GET /jobs/jobId/things?status=status&maxResults=maxResults&nextToken=nextToken
```

Para obter mais informações, consulte`ListJobExecutionsForJob`

CLI syntax

```
aws iot  list-job-executions-for-job \
--job-id <value> \
[--status <value>] \
[--max-results <value>] \
[--next-token <value>]  \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

`cli-input-json` format:

```
{
"jobId": "string",
"status": "string",
"maxResults": "integer",
"nextToken": "string"
}
```

Para obter mais informações, consulte`list-job-executions-for-job`

ListJobExecutionsForThing

Obtém uma lista de execuções de trabalhos para uma coisa.

HTTPS request

```
GET /things/thingName/jobs?status=status&maxResults=maxResults&nextToken=nextToken
```

Para obter mais informações, consulte`ListJobExecutionsForThing`

CLI syntax

```
aws iot list-job-executions-for-thing \
--thing-name <value> \
[--status <value>] \
```

```
[--max-results <value>] \
[--next-token <value>]  \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

`cli-input-json` format:

```
{
"thingName": "string",
"status": "string",
"maxResults": "integer",
"nextToken": "string"
}
```

Para obter mais informações, consulte`list-job-executions-for-thing`

ListJobs

Obtém uma lista de trabalhos noConta da AWS.

HTTPS request

```
GET /jobs?
status=status&targetSelection=targetSelection&thingGroupName=thingGroupName&thingGroupId=thingGroup
```

Para obter mais informações, consulte`ListJobs`

CLI syntax

```
aws iot list-jobs \
[--status <value>] \
[--target-selection <value>] \
[--max-results <value>] \
[--next-token <value>] \
[--thing-group-name <value>] \
[--thing-group-id <value>]  \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

`cli-input-json` format:

```
{
"status": "string",
"targetSelection": "string",
"maxResults": "integer",
"nextToken": "string",
"thingGroupName": "string",
"thingGroupId": "string"
}
```

Para obter mais informações, consulte`list-jobs`

UpdateJob

Atualiza campos compatíveis do trabalho especificado. Valores atualizados para `timeoutConfig` entrar em vigor somente para novas execuções em andamento. Atualmente, as execuções em andamento continuam a ter a antiga configuração de tempo limite.

HTTPS request

```
PATCH /jobs/jobId
{
"description": "string",
"presignedUrlConfig": {
  "expiresInSec": number,
  "roleArn": "string"
},
"jobExecutionsRolloutConfig": {
  "exponentialRate": {
     "baseRatePerMinute": number,
     "incrementFactor": number,
     "rateIncreaseCriteria": {
        "numberOfNotifiedThings": number,
        "numberOfSucceededThings": number
     },
  "maximumPerMinute": number
  },
"abortConfig": {
  "criteriaList": [
     {
        "action": "string",
        "failureType": "string",
        "minNumberOfExecutedThings": number,
        "thresholdPercentage": number
     }
  ]
},
"timeoutConfig": {
  "inProgressTimeoutInMinutes": number
}
}
```

Para obter mais informações, consulte`UpdateJob`

CLI syntax

```
aws iot  update-job \
--job-id <value> \
[--description <value>] \
[--presigned-url-config <value>] \
[--job-executions-rollout-config <value>] \
[--abort-config <value>] \
[--timeout-config <value>] \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

`cli-input-json` format:

```
{
"description": "string",
"presignedUrlConfig": {
  "expiresInSec": number,
  "roleArn": "string"
},
"jobExecutionsRolloutConfig": {
  "exponentialRate": {
     "baseRatePerMinute": number,
     "incrementFactor": number,
     "rateIncreaseCriteria": {
        "numberOfNotifiedThings": number,
        "numberOfSucceededThings": number
```

```
      }
    },
    "maximumPerMinute": number
  },
  "abortConfig": {
    "criteriaList": [
      {
        "action": "string",
        "failureType": "string",
        "minNumberOfExecutedThings": number,
        "thresholdPercentage": number
      }
    ]
  },
  "timeoutConfig": {
    "inProgressTimeoutInMinutes": number
  }
}
```

Para obter mais informações, consulte`update-job`

APIs e tipos de dados MQTT e HTTPS de dispositivos de trabalho

Os seguintes comandos estão disponíveis por meio dos protocolos MQTT e HTTPS. Use essas operações de API no plano de dados para dispositivos que executam as tarefas.

Tipos de dados MQTT e HTTPS de dispositivos de trabalho

Os seguintes tipos de dados são usados para comunicação com o serviço Jobs da AWS IoT por meio dos protocolos MQTT e HTTPS.

JobExecution

Contém dados sobre a execução de um trabalho. O exemplo a seguir mostra a sintaxe:

```
{
    "jobId" : "string",
    "thingName" : "string",
    "jobDocument" : "string",
    "status": "QUEUED|IN_PROGRESS|FAILED|SUCCEEDED|CANCELED|TIMED_OUT|REJECTED|REMOVED",
    "statusDetails": {
        "string": "string"
    },
    "queuedAt" : "timestamp",
    "startedAt" : "timestamp",
    "lastUpdatedAt" : "timestamp",
    "versionNumber" : "number",
    "executionNumber": long
}
```

Para obter mais informações, consulte `JobExecution` ou `job-execution`.

JobExecutionState

Contém dados sobre o estado de uma execução de trabalho. O exemplo a seguir mostra a sintaxe:

```
{
```

```
    "status": "QUEUED|IN_PROGRESS|FAILED|SUCCEEDED|CANCELED|TIMED_OUT|REJECTED|REMOVED",
    "statusDetails": {
        "string": "string"
        ...
    }
    "versionNumber": "number"
}
```

Para obter mais informações, consulte `JobExecutionState` ou `job-execution-state`.

JobExecutionSummary

Contém um subconjunto de informações sobre a execução de um trabalho. O exemplo a seguir mostra a sintaxe:

```
{
    "jobId": "string",
    "queuedAt": timestamp,
    "startedAt": timestamp,
    "lastUpdatedAt": timestamp,
    "versionNumber": "number",
    "executionNumber": long
}
```

Para obter mais informações, consulte `JobExecutionSummary` ou `job-execution-summary`.

Saiba mais sobre as operações da API MQTT e HTTPS nas seguintes seções:

- API MQTT de dispositivos de trabalho (p. 756)
- API HTTP do dispositivo de trabalho (p. 762)

API MQTT de dispositivos de trabalho

Você pode emitir comandos de dispositivo jobs publicando mensagens MQTT naTópicos reservados usados para comandos de trabalhos (p. 109).

Seu cliente do lado do dispositivo deve estar inscrito nos tópicos da mensagem de resposta desses comandos. Se você usar oAWS IoTDevice Client, seu dispositivo assinará automaticamente os tópicos de resposta. Isso significa que o corretor de mensagens publicará tópicos de mensagem de resposta para o cliente que publicou a mensagem de comando, independentemente de o cliente se inscrever nos tópicos da mensagem de resposta ou não. Essas mensagens de resposta não passam pelo corretor de mensagens e não podem ser inscritas por outros clientes ou regras.

Ao se inscrever no trabalho e`jobExecution`tópicos de eventos para sua solução de monitoramento de frotas, primeiro ativeeventos de execução de trabalho e trabalho (p. 1107)para receber eventos no lado da nuvem. Mensagens de andamento do Job que são processadas por meio do corretor de mensagens e podem ser usadas porAWS IoTas regras são publicadas comoEventos de trabalho (p. 1118). Como o corretor de mensagens publica mensagens de resposta, mesmo sem uma assinatura explícita para elas, seu cliente deve estar configurado para receber e identificar as mensagens que recebe. Seu cliente também deve confirmar que o*`thingName`*no tópico da mensagem recebida se aplica ao nome da coisa do cliente antes que o cliente atue na mensagem.

> **Note**
>
> Mensagens queAWS IoTenvia em resposta às mensagens de comando da MQTT Jobs API são cobradas em sua conta, independentemente de você se inscrever explicitamente ou não.

A seguir, mostra as operações da API MQTT e sua sintaxe de solicitação e resposta. Todas as operações da API MQTT têm os seguintes parâmetros:

clientToken

Um token do cliente opcional que pode ser usado para correlacionar solicitações e respostas. Insira um valor arbitrário aqui e ele será refletido na resposta.

`timestamp`

O tempo em segundos, desde oépoca, quando a mensagem foi enviada.

GetPendingJobExecutions

Obtém a lista de todos os trabalhos que não estão em um status final, para uma coisa especificada.

Para invocar essa API, publique uma mensagem em `$aws/things/thingName/jobs/get`.

Carga da solicitação:

```
{ "clientToken": "string" }
```

O corretor de mensagens publicará`$aws/things/thingName/jobs/get/accepted`e`$aws/things/thingName/jobs/get/rejected`mesmo sem uma assinatura específica para eles. No entanto, para que seu cliente receba as mensagens, ele deve estar ouvindo. Para obter mais informações, consultea nota sobre mensagens da Jobs API (p. 756).

Carga da resposta:

```
{
"inProgressJobs" : [ JobExecutionSummary ... ],
"queuedJobs" : [ JobExecutionSummary ... ],
"timestamp" : 1489096425069,
"clientToken" : "client-001"
}
```

Onde`inProgressJobs`e`queuedJobs`retorna uma lista deJobExecutionSummary (p. 756)objetos que têm status de`IN_PROGRESS`ou`QUEUED`.

StartNextPendingJobExecution

Obtém e inicia a próxima execução de trabalho pendente de uma coisa (status`IN_PROGRESS`ou`QUEUED`).

- Todas as execuções de trabalho com status`IN_PROGRESS`são retornados primeiro.
- As execuções de trabalho são retornadas na ordem em que foram criadas.
- Se a próxima execução de trabalho pendente for`QUEUED`, seu estado muda para`IN_PROGRESS`e os detalhes de status da execução do trabalho são definidos conforme especificado.
- Se a próxima execução de trabalho pendente já estiver`IN_PROGRESS`, seus detalhes de status não são alterados.
- Se nenhuma execução de trabalho estiver pendente, a resposta não incluirá o campo `execution`.
- Você pode criar um temporizador de etapa. Para isso, defina um valor para `stepTimeoutInMinutes`propriedade. Se você não atualizar o valor dessa propriedade executando `UpdateJobExecution`, a execução do trabalho atingirá o tempo limite quando o temporizador de etapa expirar.

Para invocar essa API, publique uma mensagem em `$aws/things/thingName/jobs/start-next`.

Carga da solicitação:

```
{
```

```
"statusDetails": {
    "string": "job-execution-state"
    ...
},
"stepTimeoutInMinutes": long,
"clientToken": "string"
}
```

`statusDetails`

Uma coleção de pares nome e valor que descrevem o status da execução do trabalho. Se não especificado, o `statusDetails` não será alterado.

`stepTimeOutInMinutes`

Especifica o tempo que este dispositivo tem para concluir a execução do trabalho. Se o status de execução do trabalho não estiver definido como um estado terminal antes que o temporizador expire ou seja redefinido (ao chamar `UpdateJobExecution`, definir o status como `IN_PROGRESS` e especificar um novo valor de tempo limite no campo `stepTimeoutInMinutes`), o status de execução do trabalho será definido como `TIMED_OUT`. A configuração do tempo limite não tem efeito sobre o tempo limite de execução desse trabalho, que pode ter sido especificado quando o trabalho foi criado (`CreateJob` usando o campo `timeoutConfig`).

O corretor de mensagens publicará`$aws/things/thingName/jobs/start-next/accepted`e`$aws/things/thingName/jobs/start-next/rejected`mesmo sem uma assinatura específica para eles. No entanto, para que seu cliente receba as mensagens, ele deve estar ouvindo. Para obter mais informações, consultea nota sobre mensagens da Jobs API (p. 756).

Carga da resposta:

```
{
"execution" : JobExecutionData,
"timestamp" : timestamp,
"clientToken" : "string"
}
```

Onde`execution`é umJobExecution (p. 755)objeto. Por exemplo:

```
{
"execution" : {
    "jobId" : "022",
    "thingName" : "MyThing",
    "jobDocument" : "< contents of job document >",
    "status" : "IN_PROGRESS",
    "queuedAt" : 1489096123309,
    "lastUpdatedAt" : 1489096123309,
    "versionNumber" : 1,
    "executionNumber" : 1234567890
},
"clientToken" : "client-1",
"timestamp" : 1489088524284,
}
```

DescribeJobExecution

Obtém informações detalhadas sobre uma execução de trabalho.

É possível definir o`jobId`para`$next`Para retornar a próxima execução de trabalho pendente para uma coisa (com um status de`IN_PROGRESS`ou`QUEUED`).

Para invocar essa API, publique uma mensagem em `$aws/things/thingName/jobs/jobId/get`.

Carga da solicitação:

```
{
"executionNumber": long,
"includeJobDocument": boolean,
"clientToken": "string"
}
```

`thingName`

O nome da coisa associada ao dispositivo.

`jobId`

O identificador exclusivo atribuído a este trabalho quando ele foi criado.

Ou use`$next`Para retornar a próxima execução de trabalho pendente para uma coisa (com um status de`IN_PROGRESS`ou`QUEUED`). Nesse caso, todas as execuções de trabalho com status`IN_PROGRESS`são retornados primeiro. As execuções de trabalho são retornadas na ordem em que foram criadas.

`executionNumber`

(Opcional) Um número que identifica a execução de um trabalho em um dispositivo. Se não especificado, a execução do trabalho mais recente será retornada.

`includeJobDocument`

(Opcional) A menos que esteja definido como`false`, a resposta conterá o documento de trabalho. O padrão é `true`.

O corretor de mensagens publicará`$aws/things/thingName/jobs/jobId/get/accepted`e`$aws/things/thingName/jobs/jobId/get/rejected`mesmo sem uma assinatura específica para eles. No entanto, para que seu cliente receba as mensagens, ele deve estar ouvindo. Para obter mais informações, consultea nota sobre mensagens da Jobs API (p. 756).

Carga da resposta:

```
{
"execution" : JobExecutionData,
"timestamp": "timestamp",
"clientToken": "string"
}
```

Onde`execution`é umJobExecution (p. 755)objeto.

UpdateJobExecution

Atualiza o status de uma execução de trabalho. Você pode criar um temporizador de etapa. Para isso, defina um valor para a propriedade `stepTimeoutInMinutes`. Se você não atualizar o valor dessa propriedade executando `UpdateJobExecution` novamente, a execução do trabalho atingirá o tempo limite quando o temporizador de etapa expirar.

Para invocar essa API, publique uma mensagem em `$aws/things/thingName/jobs/jobId/update`.

Carga da solicitação:

```
{
```

```
"status": "job-execution-state",
"statusDetails": {
    "string": "string"
    ...
},
"expectedVersion": "number",
"executionNumber": long,
"includeJobExecutionState": boolean,
"includeJobDocument": boolean,
"stepTimeoutInMinutes": long,
"clientToken": "string"
}
```

`status`

O novo status para a execução do trabalho (`IN_PROGRESS`,`FAILED`,`SUCCEEDED`, ou`REJECTED`). Isso deve ser especificado em todas as atualizações.

`statusDetails`

Uma coleção de pares nome e valor que descrevem o status da execução do trabalho. Se não especificado, o `statusDetails` não será alterado.

`expectedVersion`

A versão esperada atual da execução do trabalho. Cada vez que você atualiza a execução do trabalho, sua versão é incrementada. Se a versão da execução do trabalho armazenada no serviço Jobs da AWS IoT não corresponder, a atualização será rejeitada com um erro `VersionMismatch`, e uma ErrorResponse (p. 740) que contém o status da execução do trabalho atual será retornada. Isso torna desnecessário executar uma solicitação `DescribeJobExecution` separada para obter os dados do status da execução do trabalho.

`executionNumber`

(Opcional) Um número que identifica a execução de um trabalho em um dispositivo. Se não especificado, a execução do trabalho mais recente será usada.

`includeJobExecutionState`

(Opcional) Quando incluído e definido como`true`, a resposta conterá o`JobExecutionState`campo. O padrão é `false`.

`includeJobDocument`

(Opcional) Quando incluído e definido como`true`, a resposta conterá o`JobDocument`. O padrão é `false`.

`stepTimeoutInMinutes`

Especifica o tempo que este dispositivo tem para concluir a execução do trabalho. Se o status de execução do trabalho não estiver definido como estado terminal antes que o temporizador expire ou seja redefinido (ao chamar `UpdateJobExecution` novamente, definir o status como `IN_PROGRESS` e especificar um novo valor de tempo limite neste campo), o status de execução do trabalho será definido como `TIMED_OUT`. A configuração ou a reconfiguração do tempo limite não tem efeito sobre o tempo limite de execução do trabalho que pode ter sido especificado quando o trabalho foi criado (usando `CreateJob` com `timeoutConfig`).

O corretor de mensagens publicará`$aws/things/thingName/jobs/jobId/update/acceptede$aws/things/thingName/jobs/jobId/update/rejected`mesmo sem uma assinatura específica para eles. No entanto, para que seu cliente receba as mensagens, ele deve estar ouvindo. Para obter mais informações, consultea nota sobre mensagens da Jobs API (p. 756).

Carga da resposta:

```
{
"executionState": JobExecutionState,
"jobDocument": "string",
"timestamp": timestamp,
"clientToken": "string"
}
```

`executionState`

Um objeto JobExecutionState (p. 755).

`jobDocument`

Um objeto documento de trabalho (p. 682).

`timestamp`

O tempo em segundos, desde oépoca, quando a mensagem foi enviada.

`clientToken`

Um token do cliente usado para correlacionar solicitações e respostas.

Ao usar o protocolo MQTT, você também pode realizar as seguintes atualizações:

JobExecutionsChanged

Enviado sempre que uma execução de trabalho é adicionada ou removida da lista de execuções de trabalhos pendentes para uma coisa.

Use o tópico :

`$aws/things/thingName/jobs/notify`

Carga da mensagem:

```
{
"jobs" : {
    "JobExecutionState": [ JobExecutionSummary ... ]
         },
    "timestamp": timestamp
}
```

NextJobExecutionChanged

Enviado sempre que houver uma alteração em qual execução de trabalho é a seguinte na lista de execuções de trabalhos pendentes para uma coisa, conforme definido para`DescribeJobExecution`com`jobId` `$next`. Essa mensagem não é enviada quando os detalhes da próxima execução de trabalho é alterada, apenas quando o próximo trabalho que seria retornado por`DescribeJobExecution`com`jobId` `$next`mudou. Considere as execuções de trabalho J1 e J2 com um status de`QUEUED`. J1 é a próxima na lista de execuções de trabalhos pendentes. Se o status de J2 for alterado para`IN_PROGRESS`Enquanto o estado de J1 permanecer inalterado, essa notificação é enviada e contém os detalhes do J2.

Use o tópico :

`$aws/things/thingName/jobs/notify-next`

Carga da mensagem:

```
{
"execution" : JobExecution,
"timestamp": timestamp,
}
```

API HTTP do dispositivo de trabalho

Os dispositivos podem se comunicar comAWS IoTJobs usando HTTP Signature Versão 4 na porta 443. Este é o método usado peloAWSSDKs e CLI. Para obter mais informações sobre essas ferramentas, consulteAWS CLIReferência de Comando:IOT-Jobs-dataouAWSSDKs e ferramentas.

Os comandos a seguir estão disponíveis para dispositivos que executam as tarefas. Para obter informações sobre o uso das operações da API do protocolo MQTT, consulteAPI MQTT de dispositivos de trabalho (p. 756).

GetPendingJobExecutions

Obtém a lista de todos os trabalhos que não estão em um status final, para uma coisa especificada.

HTTPS request

```
GET /things/thingName/jobs
```

Resposta:

```
{
"inProgressJobs" : [ JobExecutionSummary ... ],
"queuedJobs" : [ JobExecutionSummary ... ]
}
```

Para obter mais informações, consulte`GetPendingJobExecutions`

CLI syntax

```
aws iot-jobs-data get-pending-job-executions \
--thing-name <value>  \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

`cli-input-json` format:

```
{
"thingName": "string"
}
```

Para obter mais informações, consulte`get-pending-job-executions`

StartNextPendingJobExecution

Obtém e inicia a próxima execução de trabalho pendente para uma coisa (com um status de`IN_PROGRESS`ou`QUEUED`).

- Todas as execuções de trabalho com status`IN_PROGRESS`são retornados primeiro.
- As execuções de trabalho são retornadas na ordem em que foram criadas.

- Se a próxima execução de trabalho pendente for`QUEUED`, seu status muda para`IN_PROGRESS`e os detalhes de status da execução do trabalho são definidos conforme especificado.
- Se a próxima execução de trabalho pendente já estiver`IN_PROGRESS`, seus detalhes de status não mudam.
- Se nenhuma execução de trabalho estiver pendente, a resposta não incluirá o campo `execution`.
- Você pode criar um temporizador de etapa. Para isso, defina um valor para `ostepTimeoutInMinutes`propriedade. Se você não atualizar o valor dessa propriedade executando `UpdateJobExecution`, a execução do trabalho atingirá o tempo limite quando o temporizador de etapa expirar.

HTTPS request

O exemplo a seguir mostra a sintaxe da solicitação:

```
PUT /things/thingName/jobs/$next
{
"statusDetails": {
    "string": "string"
    ...
},
"stepTimeoutInMinutes": long
}
```

Para obter mais informações, consulte`StartNextPendingJobExecution`

CLI syntax

Resumo:

```
aws iot-jobs-data  start-next-pending-job-execution \
--thing-name <value> \
{--step-timeout-in-minutes <value>] \
[--status-details <value>]  \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

`cli-input-json` format:

```
{
"thingName": "string",
"statusDetails": {
"string": "string"
},
"stepTimeoutInMinutes": long
}
```

Para obter mais informações, consulte`start-next-pending-job-execution`

DescribeJobExecution

Obtém informações detalhadas sobre uma execução de trabalho.

É possível definir `ojobId`para`$next`para retornar a próxima execução de trabalho pendente para uma coisa. O status da execução do trabalho deve ser `QUEUED` ou `IN_PROGRESS`.

HTTPS request

Solicitação:

```
GET /things/thingName/jobs/jobId?
executionNumber=executionNumber&includeJobDocument=includeJobDocument
```

Resposta:

```
{
"execution" : JobExecution,
}
```

Para obter mais informações, consulte`DescribeJobExecution`

CLI syntax

Resumo:

```
aws iot-jobs-data  describe-job-execution \
--job-id <value> \
--thing-name <value> \
[--include-job-document | --no-include-job-document] \
[--execution-number <value>]  \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

`cli-input-json` format:

```
{
"jobId": "string",
"thingName": "string",
"includeJobDocument": boolean,
"executionNumber": long
}
```

Para obter mais informações, consulte`describe-job-execution`

UpdateJobExecution

Atualiza o status de uma execução de trabalho. Você pode criar um temporizador de etapa. Para isso, defina um valor para `ostepTimeoutInMinutes`propriedade. Se você não atualizar o valor dessa propriedade executando `UpdateJobExecution` novamente, a execução do trabalho atingirá o tempo limite quando o temporizador de etapa expirar.

HTTPS request

Solicitação:

```
POST /things/thingName/jobs/jobId
{
"status": "job-execution-state",
"statusDetails": {
    "string": "string"
    ...
},
"expectedVersion": "number",
"includeJobExecutionState": boolean,
"includeJobDocument": boolean,
"stepTimeoutInMinutes": long,
"executionNumber": long
}
```

Para obter mais informações, consulte`UpdateJobExecution`

CLI syntax

Resumo:

```
aws iot-jobs-data  update-job-execution \
--job-id <value> \
--thing-name <value> \
--status <value> \
[--status-details <value>] \
[--expected-version <value>] \
[--include-job-execution-state | --no-include-job-execution-state] \
[--include-job-document | --no-include-job-document] \
[--execution-number <value>]  \
[--cli-input-json <value>] \
[--step-timeout-in-minutes <value>] \
[--generate-cli-skeleton]
```

`cli-input-json` format:

```
{
"jobId": "string",
"thingName": "string",
"status": "string",
"statusDetails": {
"string": "string"
},
"stepTimeoutInMinutes": number,
"expectedVersion": long,
"includeJobExecutionState": boolean,
"includeJobDocument": boolean,
"executionNumber": long
}
```

Para obter mais informações, consulte `update-job-execution`.

Protegendo usuários e dispositivos comAWS IoTTrabalhos do

Para autorizar os usuários a usarAWS IoTTrabalhos com seus dispositivos, você deve conceder permissões a eles usando políticas do IAM. Os dispositivos devem então ser autorizados usandoAWS IoT Corepolíticas para se conectar com segurançaAWS IoT, receba execuções de trabalhos e atualize o status da execução.

Tipo de política do necessárioAWS IoTTrabalhos do

A tabela a seguir mostra os diferentes tipos de políticas que você deve usar para autorização. Para obter mais informações sobre a política necessária para usar, consulteAutorização (p. 330).

Tipo de política necessário

Caso de uso	Protocolo	Autenticação	Plano de controlo/plano de dados	Tipo de identidade	Tipo de política necessário
Autorizar um administrador, operador	HTTPS	AWSAutenticação Signature versão 4 (porta 443)	Plano de controle e plano de dados	Amazon Cognito Identity, IAM	Política do IAM

Caso de uso	Protocolo	Autenticação	Plano de controlo/plano de dados	Tipo de identidade	Tipo de política necessário
ou serviço em nuvem a trabalhar com segurança com Jobs				ou usuário federado	
Autorize seu dispositivo IoT a trabalhar com segurança com Jobs	MQTT/ HTTPS	Autenticação mútua TCP ou TLS (porta 8883 ou 443)	Plano de dados	Certificados X.509	Política do AWS IoT Core

Para autorizarAWS IoTOperações de tarefas que podem ser executadas tanto no plano de controle quanto no plano de dados, você deve usar políticas do IAM. As identidades devem ter sido autenticadas comAWS IoTpara executar essas operações, que devem serIdentidades do Amazon Cognito (p. 318)ouUsuários, grupos e funções do IAM (p. 317). Para obter mais informações sobre a autenticação, consulte Autenticação (p. 294).

Os dispositivos agora devem ser autorizados no plano de dados usandoAWS IoT Corepolíticas para se conectar com segurança ao gateway do dispositivo. O gateway do dispositivo permite que os dispositivos se comuniquem com segurança comAWS IoT, receba execuções de trabalhos e atualize o status da execução do trabalho. A comunicação do dispositivo é protegida usando o seguroMQTT (p. 86)ouHTTPS (p. 97)protocolos de comunicação. Esses protocolos usamCertificados do cliente X.509 (p. 298)que são fornecidos porAWS IoTpara autenticar as conexões do dispositivo.

A seguir, mostra como você autoriza seus usuários, cServiços altose dispositivos para usarAWS IoTTrabalhos do. Para obter informações sobre as operações de API de plano de controle e plano de dados da, consulteAWS IoTAPIs de trabalhos (p. 740).

Tópicos

- Autorizar usuários e serviços em nuvem a usarAWS IoTTrabalhos do (p. 766)
- Autorizar seus dispositivos a usar com segurançaAWS IoTTrabalhos no plano de dados (p. 772)

Autorizar usuários e serviços em nuvem a usarAWS IoTTrabalhos do

Para autorizar seus usuários e serviços em nuvem, você deve usar as políticas do IAM no plano de controle e no plano de dados. As políticas devem ser usadas com o protocolo HTTPS e devem usarAWSAssinatura versão 4 (porta 443) para autenticar usuários.

> **Note**
>
> AWS IoT Corepolíticas não devem ser usadas no plano de controle. Somente as políticas do IAM são usadas para autorizar usuários ou Cloud Services. Para obter mais informações sobre como usar o tipo de política necessário, consulteTipo de política do necessárioAWS IoTTrabalhos do (p. 765).

As políticas do IAM são documentos JSON que contêm declarações de política. Uso de declarações de políticasEfeito,Ação, eRecursoElementos para especificar recursos, ações permitidas ou negadas e condições sob as quais as ações são permitidas ou negadas. Para obter mais informações, consulte Referência de elementos de política JSON do IAM no Guia do usuário do IAM.

> **Warning**
>
> Recomendamos não usar permissões curinga, como`"Action": ["iot:*"]`em suas políticas do IAM ouAWS IoT Corepolíticas. Usar permissões curinga não é uma melhor prática de segurança recomendada. Para obter mais informações, consulte AWS IoTPolíticas excessivamente permissivas (p. 948).

Políticas do IAM no plano de controle

No plano de controle, as políticas do IAM usam `oiot:`prefixo com a ação para autorizar a operação da API de trabalhos correspondentes. Por exemplo, `oiot:CreateJob`Concede permissão de política ao usuário do para usar o`CreateJob`API.

Ações de políticas

A tabela a seguir mostra uma lista de ações e permissões de política do IAM para usar as ações da API. Para obter informações sobre tipos de recursos, consulteTipos de recursos definidos porAWS IoT. Para obter mais informações sobreAWS IoTações, consulteAções definidas peloAWS IoT.

Ações de política do IAM no plano de controle

Ação de política	Operação de API	Tipos de recursos	Descrição
`iot:AssociateTargetsWithJob`	`AssociateTargetsWithJob`	• trabalho • thing • thingGroups	Representa a permissão para associar um grupo a um trabalho contínuo. O`iot:AssociateTargetsWithJob`A permissão é verificada sempre que é feita uma solicitação para associar destinos.
`iot:CancelJob`	`CancelJob`	trabalho	Representa a permissão para cancelar um trabalho. O`iot:CancelJob`A permissão é verificada sempre que é feita uma solicitação para cancelar um trabalho.
`iot:CancelJobExecution`	`CancelJobExecution`	• trabalho • thing	Representa a permissão para cancelar uma execução de tarefa. O`iot: CancelJobExecution`A permissão é verificada sempre que é feita uma solicitação para cancelar uma execução de tarefa.
`iot:CreateJob`	`CreateJob`	• trabalho • thing • thingGroups • modelo de trabalho	Representa a permissão para criar um trabalho. O`iot: CreateJob`A permissão é verificada sempre que é feita uma solicitação para criar um trabalho.
`iot:CreateJobTemplate`	`CreateJobTemplate`	• trabalho • modelo de trabalho	Representa a permissão para criar um modelo de trabalho. O`iot: CreateJobTemplate`A permissão é verificada sempre que é feita uma solicitação para criar um modelo de trabalho.
`iot:DeleteJob`	`DeleteJob`	trabalho	Representa a permissão para excluir um trabalho. O`iot: DeleteJob`A permissão é verificada sempre que é feita uma solicitação para excluir um trabalho.

Ação de política	Operação de API	Tipos de recursos	Descrição
`iot:DeleteJobTemplate`	`DeleteJobTemplate`	modelo de trabalho	Representa a permissão para excluir um modelo de trabalho. O`iot: CreateJobTemplate`A permissão é verificada sempre que é feita uma solicitação para excluir um modelo de trabalho.
`iot:DeleteJobExecution`	`DeleteJobTemplate`	• trabalho • thing	Representa a permissão para excluir a execução de um trabalho. O`iot: DeleteJobExecution`A permissão é verificada sempre que é feita uma solicitação para excluir uma execução de tarefa.
`iot:DescribeJob`	`DescribeJob`	trabalho	Representa a permissão para descrever um trabalho. O`iot: DescribeJob`A permissão é verificada sempre que é feita uma solicitação para descrever um trabalho.
`iot:DescribeJobExecution`	`DescribeJobExecution`	• trabalho • thing	Representa a permissão para descrever a execução de um trabalho. O`iot: DescribeJobExecution`A permissão é verificada sempre que é feita uma solicitação para descrever a execução de um trabalho.
`iot:DescribeJobTemplate`	`DescribeJobTemplate`	modelo de trabalho	Representa a permissão para descrever um modelo de trabalho. O`iot: DescribeJobTemplate`A permissão é verificada sempre que é feita uma solicitação para descrever um modelo de trabalho.
`iot:DescribeManagedJobTemplate`	`DescribeManagedJobTemplate`	modelo de trabalho	Representa a permissão para descrever um modelo de trabalho gerenciado. O`iot: DescribeManagedJobTemplate`A permissão é verificada sempre que é feita uma solicitação para descrever um modelo de trabalho gerenciado.
`iot:GetJobDocument`	`GetJobDocument`	trabalho	Representa a permissão para obter o documento de trabalho para um trabalho. O`iot:GetJobDocument`A permissão é verificada sempre que é feita uma solicitação para obter um documento de trabalho.
`iot:ListJobExecutionsForJob`	`ListJobExecutionsForJob`	trabalho	Representa a permissão para listar as execuções de um trabalho. O`iot:ListJobExecutionsForJob`A permissão é verificada sempre que é feita uma solicitação para listar as execuções de um trabalho.
`iot:ListJobExecutionsForThing`	`ListJobExecutionsForThing`	thing	Representa a permissão para listar as execuções de um trabalho. O`iot:ListJobExecutionsForThing`A permissão é verificada sempre que é feita uma solicitação para listar as execuções de um negócio.

Ação de política	Operação de API	Tipos de recursos	Descrição
`iot:ListJobs`	`ListJobs`	none	Representa a permissão para listar os trabalhos. O`iot:ListJobs`A permissão é verificada sempre que é feita uma solicitação para listar os trabalhos.
`iot:ListJobTemplates`	`ListJobTemplates`	none	Representa a permissão para listar os modelos de trabalho. O`iot:ListJobTemplates`A permissão é verificada sempre que é feita uma solicitação para listar os modelos de trabalho.
`iot:ListManagedJobTemplates`	`ListManagedJobTemplates`	none	Representa a permissão para listar os modelos de trabalho gerenciados. O`iot:ListManagedJobTemplates`A permissão é verificada sempre que é feita uma solicitação para listar os modelos de trabalho gerenciados.
`iot:UpdateJob`	`UpdateJob`	trabalho	Representa a permissão para atualizar um trabalho. O`iot:UpdateJob`A permissão é verificada sempre que é feita uma solicitação para atualizar um trabalho.
`iot:TagResource`	`TagResource`	• trabalho • modelo de trabalho • thing	Concede permissão para marcar um recurso específico.
`iot:UntagResource`	`UntagResource`	• trabalho • modelo de trabalho • thing	Concede permissão para desmarcar um recurso específico.

Exemplo de política básica do IAM

A seguir, mostra um exemplo de uma política do IAM que permite ao usuário a permissão para executar as seguintes ações para o grupo de coisas e coisas da IoT.

No exemplo, substitua:

- *`região`*Com suas necessidadesRegião da AWS, por exemplo,`us-east-1`.
- *`account-id`*Com suas necessidadesConta da AWSnúmero, como`57EXAMPLE833`.
- *`coisa-nome do grupo`*com o nome do seu grupo de coisas de IoT para o qual você está segmentando trabalhos, como`FirmwareUpdateGroup`.
- *`Nome da coisa`*com o nome da sua coisa de IoT para a qual você está segmentando trabalhos, como`MyIoTThing`.

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

```
        {
            "Action": [
                "iot:CreateJobTemplate",
                "iot:CreateJob",
                ],
            "Effect": "Allow",
            "Resource": "arn:aws:iot:region:account-id:thinggroup/thing-group-name"
        },
        {
            "Action": [
                "iot:DescribeJob",
                "iot:CancelJob",
                "iot:DeleteJob",
            ],
            "Effect": "Allow",
            "Resource": "arn:aws:iot:region:account-id:job/*"
        },
        {
            "Action": [
                "iot:DescribeJobExecution",
                "iot:CancelJobExecution",
                "iot:DeleteJobExecution",
            ],
            "Effect": "Allow",
            "Resource": [
                "arn:aws:iot:region:account-id:thing/thing-name"
                "arn:aws:iot:region:account-id:job/*"
            ]
        }
    ]
}
```

Políticas do IAM no plano de dados

As políticas do IAM no plano de dados usam `oiotjobsdata:`prefixo para autorizar operações de API de trabalhos que os usuários podem executar. No plano de dados, você pode conceder a um usuário permissão para usar o`DescribeJobExecution`API do usando `oiotjobsdata:DescribeJobExecution`Ação de política.

> **Warning**
>
> O uso de políticas do IAM no plano de dados não é recomendado ao segmentarAWS IoTTrabalhos para seus dispositivos. Recomendamos que você use políticas do IAM no plano de controle para que os usuários criem e gerenciem trabalhos. No plano de dados, para autorizar dispositivos a recuperar execuções de tarefas e atualizar o status da execução, useAWS IoT CorePolíticas para protocolo HTTPS (p. 773).

Exemplo de política básica do IAM

Geralmente, as operações de API que devem ser autorizadas são executadas por um usuário digitando comandos da CLI da. Veja a seguir um exemplo de um usuário que executa um`DescribeJobExecution`operação.

No exemplo, substitua:

- *`região`*Com suas necessidadesRegião da AWS, por exemplo,`us-east-1`.
- *`account-id`*Com suas necessidadesConta da AWSnúmero, como`57EXAMPLE833`.
- *`Nome da coisa`*com o nome da sua coisa de IoT para a qual você está segmentando trabalhos, como`myRegisteredThing`.
- *`job-id`*é o identificador exclusivo do trabalho que é direcionado usando a API.

```
aws iot-jobs-data describe-job-execution \
    --endpoint-url "https://account-id.jobs.iot.region.amazonaws.com" \
    --job-id jobID --thing-name thing-name
```

Veja a seguir um exemplo da política IAM que autoriza essa ação:

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

Exemplo de política IAM para plano de controle e plano de dados

Se um usuário executar uma operação de API no plano de controle e no plano de dados, sua ação de política de plano de controle deverá usar `oiot:`prefixo, e sua ação de política de plano de dados deve usar `oiotjobsdata:`prefixo.

Por exemplo, as receitas`DescribeJobExecution`A API pode ser usada no plano de controle e no plano de dados. No plano de controle, oDescribeJobExecutionA API é usada para descrever a execução de um trabalho. No plano de dados, oDescribeJobExecutionA API é usada para obter detalhes de execução de um trabalho.

A política do IAM a seguir autoriza permissão de usuário para usar o`DescribeJobExecution`API do plano de controle e plano de dados.

No exemplo, substitua:

- *região*Com suas necessidadesRegião da AWS, por exemplo,`us-east-1`.
- *account-id*Com suas necessidadesConta da AWSnúmero, como`57EXAMPLE833`.
- *Nome da coisa*com o nome da sua coisa de IoT para a qual você está segmentando trabalhos, como`MyIoTThing`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": ["iotjobsdata:DescribeJobExecution"],
            "Effect": "Allow",
            "Resource": "arn:aws:iot:region:account-id:thing/thing-name"
        },
        {
            "Action": [
                "iot:DescribeJobExecution",
                "iot:CancelJobExecution",
                "iot:DeleteJobExecution",
            ],
            "Effect": "Allow",
            "Resource": [
                "arn:aws:iot:region:account-id:thing/thing-name"
                "arn:aws:iot:region:account-id:job/*"
            ]
        }
    ]
```

```
}
```

Autorizar a marcação de recursos de IoT

Para melhor controle sobre trabalhos e modelos de trabalho que um usuário pode criar, modificar ou usar, você pode anexar tags aos trabalhos ou modelos de trabalho. As tags também ajudam você a discernir a propriedade e atribuir e alocar custos colocando-os em grupos de faturamento e anexando tags a eles.

Quando um usuário deseja marcar seus trabalhos ou modelos de trabalho que ele criou usando oAWS Management Consoleou oAWS CLI, sua política IAM deve conceder permissões de usuário para marcá-los. Para conceder permissões, sua política do IAM deve usar `oiot:TagResource`Ação .

Para obter informações gerais sobre como marcar os recursos, consulteMarcar recursos do AWS IoT (p. 288).

Exemplo de política do IAM

Para obter um exemplo que mostra como conceder permissões de marcação, considere um usuário que executa o seguinte comando para criar um trabalho e marcá-lo para um ambiente específico.

No exemplo, substitua:

- *região*Com suas necessidadesRegião da AWS, por exemplo,`us-east-1`.
- *account-id*Com suas necessidadesConta da AWSnúmero, como`57EXAMPLE833`.
- *Nome da coisa*com o nome da sua coisa de IoT para a qual você está segmentando trabalhos, como`MyIoTThing`.

```
aws iot create-job
    --job-id test_job
    --targets "arn:aws:iot:region:account-id:thing/thingOne"
    --document-source "https://s3.amazonaws.com/my-s3-bucket/job-document.json"
    --description "test job description"
    --tags Key=environment,Value=beta
```

Para este exemplo, você deve usar a seguinte política IAM:

```
{
    "Version": "2012-10-17",
    "Statement":
    {
        "Action": [ "iot:CreateJob", "iot:CreateJobTemplate", "iot:TagResource" ],
        "Effect": "Allow",
        "Resource": [
            "arn:aws:iot:aws-region:account-id:job/*",
            "arn:aws:iot:aws-region:account-id:jobtemplate/*"
        ]
    }
}
```

Autorizar seus dispositivos a usar com segurançaAWS IoTTrabalhos no plano de dados

Para autorizar seus dispositivos a interagir de forma segura comAWS IoTTrabalhos no plano de dados, você deve usarAWS IoT Corepolíticas.AWS IoT Corepolíticas para trabalhos são documentos JSON que contêm declarações de política. Essas políticas também usamEfeito,Ação, eRecursoe siga uma convenção semelhante às políticas do IAM. Para obter mais informações sobre os elementos, consulteReferência de elementos de política JSON do IAMnoManual do usuário do IAM.

As políticas podem ser usadas com protocolos MQTT e HTTPS e devem usar autenticação mútua TCP ou TLS para autenticar os dispositivos. A seguir, mostra como usar essas políticas nos diferentes protocolos de comunicação.

> **Warning**
>
> Recomendamos não usar permissões curinga, como`"Action": ["iot:*"]`em suas políticas do IAM ouAWS IoT Corepolíticas. Usar permissões curinga não é uma melhor prática de segurança recomendada. Para obter mais informações, consulte AWS IoTPolíticas excessivamente permissivas (p. 948).

AWS IoT CorePolíticas para o protocolo MQTT

AWS IoT Corepolíticas para o protocolo MQTT concedem permissões para usar as ações da API MQTT do dispositivo de trabalhos. As operações da API MQTT são usadas para trabalhar com tópicos do MQTT que são reservados para comandos de jobs. Para obter mais informações sobre essas operações de API, consulteAPI MQTT de dispositivos de trabalho (p. 756).

As políticas do MQTT usam ações de política como`iot:Connect`,`iot:Publish`,`iot:Subscribe`, e`iot:Receieve`para trabalhar com os tópicos de trabalho. Essas políticas permitem que você se conecte ao agente de mensagem da, assine os tópicos do MQTT de trabalhos e envie e receba mensagens MQTT entre seus dispositivos e a nuvem. Para obter mais informações sobre essas ações, consulte Ações de políticas do AWS IoT Core (p. 333).

Para obter informações sobre tópicos paraAWS IoTEmpregos, consulteTópicos de trabalhos (p. 109).

Exemplo de política básica do MQTT

O exemplo a seguir mostra como usar`iot:Publish`e`iot:Subscribe`para publicar e assinar trabalhos e execuções de empregos.

No exemplo, substitua:

- *`região`*Com suas necessidadesRegião da AWS, por exemplo,`us-east-1`.
- *`account-id`*Com suas necessidadesConta da AWSnúmero, como`57EXAMPLE833`.
- *`Nome da coisa`*com o nome da sua coisa de IoT para a qual você está segmentando trabalhos, como`MyIoTThing`.

```
{
    "Statement": [
        {
           "Effect": "Allow",
           "Action": [
                "iot:Publish",
                "iot:Subscribe"
            ],
            "Resource": [
                "arn:aws:iot:region:account-id:topic/$aws/events/job/*",
                "arn:aws:iot:region:account-id:topic/$aws/events/jobExecution/*",
                "arn:aws:iot:region:account-id:topic/$aws/things/thing-name/jobs/*"
            ]
        }
    ],
    "Version": "2012-10-17"
}
```

AWS IoT CorePolíticas para protocolo HTTPS

AWS IoT Corepolíticas no plano de dados também podem usar o protocolo HTTPS com o mecanismo de autenticação TLS para autorizar seus dispositivos. No plano de dados, as políticas usam

`oiotjobsdata:`prefixo para autorizar operações de API de trabalhos que seus dispositivos podem executar. Por exemplo, as receitas`iotjobsdata:DescribeJobExecution`Concede permissão ao usuário do para usar o`DescribeJobExecution`API.

> **Note**
>
> As ações da política do plano de dados devem usar `oiotjobsdata:`prefixo. No plano de controle, as ações devem usar `oiot:`prefixo. Para obter um exemplo de política do IAM quando as ações de política do plano de controle e do plano de dados são usadas, consulteExemplo de política IAM para plano de controle e plano de dados (p. 771).

Ações de políticas

A tabela a seguir mostra uma lista deAWS IoT Coreações de política e permissões para autorizar dispositivos a usar as ações da API. Para obter uma lista das operações de API que você pode realizar no plano de dados, consulteAPI HTTP do dispositivo de trabalho (p. 762).

> **Note**
>
> Essas ações de política de execução de trabalhos se aplicam apenas ao endpoint HTTP TLS. Se você usar o endpoint MQTT, deverá usar as ações de política do MQTT definidas anteriormente.

AWS IoT Coreações de política no plano de dados

Ação de política	Operação de API	Tipos de recursos	Descrição
`iotjobsdata:DescribeJobExecution`	`DescribeJobExecution`	• trabalho • thing	Representa a permissão para recuperar uma execução de tarefa. O`iotjobsdata:DescribeJobExecution`A permissão é verificada sempre que é feita uma solicitação para recuperar uma execução de tarefa.
`iotjobsdata:GetPendingJobExecutions`	`GetPendingJobExecutions`	thing	Representa a permissão para recuperar a lista de trabalhos que não estão em status terminal para uma coisa. A permissão `iotjobsdata:GetPendingJobExecutions` é verificada sempre que é feita uma solicitação para recuperar a lista.
`iotjobsdata:StartNextPendingJobExecution`	`StartNextPendingJobExecution`	thing	Representa a permissão para obter e iniciar a próxima execução de trabalho pendente para uma coisa. Isto é, para atualizar uma execução de tarefa com status`QUEUED`para`IN_PROGRESS`. A permissão `iot:StartNextPendingJobExecution` é verificada sempre que é feita uma solicitação para iniciar a próxima execução de trabalho pendente.
`iotjobsdata:UpdateJobExecution`	`UpdateJobExecution`	thing	Representa a permissão para atualizar uma execução de trabalho. A permissão `iot:UpdateJobExecution` é verificada sempre que é feita uma solicitação para atualizar o estado de uma execução de trabalho.

Exemplo de política básica

Veja a seguir um exemplo de umAWS IoT Corepolítica que permite ao usuário permissão para executar as ações nas operações da API do plano de dados para qualquer recurso. Você pode definir o escopo de sua política para um recurso específico, como uma coisa de IoT. No exemplo, substitua:

- *região*Com suas necessidadesRegião da AWScomo`us-east-1`.
- *account-id*Com suas necessidadesConta da AWSnúmero, como`57EXAMPLE833`.
- *Nome da coisa*Com o nome da coisa da IoT, como`MyIoTthing`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "iotjobsdata:GetPendingJobExecutions",
                "iotjobsdata:StartNextPendingJobExecution",
                "iotjobsdata:DescribeJobExecution",
                "iotjobsdata:UpdateJobExecution"
            ],
            "Effect": "Allow",
            "Resource": "arn:aws:iot:region:account-id:thing/thing-name"
        }
    ]
}
```

Um exemplo de quando você deve usar essas políticas pode ser quando seus dispositivos IoT usam umAWS IoT Corepolítica para acessar uma dessas operações de API, como o exemplo a seguir do`DescribeJobExecution`API:

```
GET /things/thingName/jobs/jobId?
executionNumber=executionNumber&includeJobDocument=includeJobDocument&namespaceId=namespaceId
 HTTP/1.1
```

Limites de trabalhos

AWS IoTOs trabalhos têm cotas de serviço, ou limites, que correspondem ao número máximo de recursos ou operações de serviço para seuConta da AWS.

Limites de trabalho ativos e simultâneos

Esta seção ajudará você a aprender mais sobre trabalhos ativos e simultâneos e os limites que se aplicam a eles.

Trabalhos ativos e limite de trabalho ativo

Quando você cria um trabalho usando oAWS IoTConsole do ou o`CreateJob`API, o status do trabalho muda para`IN_PROGRESS`. Todos os trabalhos em andamento sãoTrabalhos ativose conte para o limite de trabalhos ativos. Isso inclui trabalhos que estão implementando novas execuções de empregos ou trabalhos que aguardam que os dispositivos concluam suas execuções de trabalho. Esse limite se aplica a trabalhos contínuos e instantâneos.

Limite de simultaneidade de trabalhos e trabalhos simultâneos

Trabalhos em andamento que estão implementando novas execuções de empregos ou trabalhos que estão cancelando execuções de tarefas criadas anteriormenteTrabalhos simultâneose conte para o

limite de simultaneidade do trabalho. ComoAWS IoTOs trabalhos podem implantar e cancelar execuções de trabalhos rapidamente a uma taxa de 1000 dispositivos por minuto, cada trabalho é`concurrente` conta para o limite de simultaneidade do trabalho apenas por um curto período de tempo. Depois que as execuções do cargo tiverem sido implementadas ou canceladas, o trabalho não é mais concorrente e não conta para o limite de simultaneidade do trabalho. Você pode usar a simultaneidade do trabalho para criar um grande número de trabalhos enquanto aguarda que os dispositivos concluam a execução do trabalho.

Para determinar se um trabalho é simultâneo, você pode usar o`IsConcurrent`Propriedade de um trabalho doAWS IoTConsole do, ou usando o`DescribeJob`ou`ListJob`API. Esse limite se aplica a trabalhos contínuos e instantâneos.

Para exibir os limites de simultaneidade de trabalhos e trabalhos ativos e outrosAWS IoTCotas de trabalhos para osConta da AWSPara solicitar um aumento de limite, consulteAWS IoTEndpoints e cotas do Gerenciamento de dispositivosnoAWSReferência geral da.

A tabela a seguir mostra os limites que se aplicam a trabalhos ativos e simultâneos.

Limites de trabalho ativos e simultâneos

Status do trabalho	Phase (Fase)	Limite de trabalhos ativos	Limite de simultaneidade de Job
`IN_PROGRESS`	Fase simultânea:AWS IoT Os trabalhos aceitam a solicitação de criação do trabalho e começam a implantar notificações de execução de tarefas em seus dispositivos. Os trabalhos nesta fase são simultâneos, conforme denotado pelo`IsConcurrent`propriedade definida como true e conta tanto para os trabalhos ativos quanto para os limites de simultaneidade do trabalho.	Aplicar	Aplicar
	Fase não concorrente:AWS IoT Os trabalhos aguardam que os dispositivos relatem os resultados de suas execuções de trabalho. Os trabalhos nesta fase contam apenas para o limite de trabalhos ativos e terão o`IsConcurrent`Propriedade definida como false.	Aplicar	Não se aplica
`Canceled`	Fase simultânea:AWS IoT Os trabalhos aceitam a solicitação de cancelamento do trabalho e começam a cancelar as execuções de tarefas criadas anteriormente para seus dispositivos. Os trabalhos nesta fase são simultâneos e terão o`IsConcurrent`Propriedade definida como true. Depois que as execuções do trabalho e do trabalho forem canceladas, o trabalho não é mais simultâneo e não conta para o limite de simultaneidade do trabalho.	Não se aplica	Aplicar

Encapsulamento seguro do AWS IoT

Quando os dispositivos são implantados atrás de firewalls restritos em sites remotos, você precisa de uma maneira de obter acesso a esses dispositivos para solução de problemas, atualizações de configuração e outras tarefas operacionais. Use encapsulamento seguro para estabelecer comunicação bidirecional com dispositivos remotos por meio de uma conexão segura gerenciada porAWS IoT. O encapsulamento seguro não requer atualizações das regras de firewall de entrada existentes para que você possa manter o mesmo nível de segurança fornecido pelas regras de firewall em um site remoto.

Por exemplo, um dispositivo de sensor localizado em uma fábrica que está a algumas centenas de milhas de distância está tendo problemas para medir a temperatura de fábrica. É possível usar o encapsulamento seguro para abrir e iniciar rapidamente uma sessão nesse dispositivo de sensor. Depois de identificar o problema (por exemplo, um arquivo de configuração incorreto), será possível redefinir o arquivo e reiniciar o dispositivo de sensor por meio da mesma sessão. Em comparação com uma solução de problemas mais tradicional (por exemplo, enviar um técnico à fábrica para investigar o dispositivo de sensor), o encapsulamento seguro diminui a resposta a incidentes e o tempo de recuperação e os custos operacionais.

O que é tunelamento seguro?

Use tunelamento seguro para acessar dispositivos que são implantados atrás de firewalls com restrição de porta em locais remotos. Você pode se conectar ao dispositivo de destino a partir de seu laptop ou computador desktop como o dispositivo de origem usando oNuvem AWS.A origem e o destino se comunicam usando um proxy local de código aberto executado em cada dispositivo. O proxy local se comunica com oNuvem AWSusando uma porta aberta permitida pelo firewall, geralmente 443. Os dados transmitidos pelo túnel são criptografados usando o Transported Layer Security (TLS).

Tópicos

- Conceitos de encapsulamento seguro (p. 777)
- Como o encapsulamento seguro do funciona (p. 778)
- Ciclo de vida do túnel seguro (p. 779)

Conceitos de encapsulamento seguro

Os termos a seguir são usados por tunelamento seguro ao estabelecer comunicação com dispositivos remotos. Para obter informações sobre como o encapsulamento seguro do funciona, consulteComo o encapsulamento seguro do funciona (p. 778).

Token de acesso para cliente (CAT)

Um par de tokens gerados pelo encapsulamento seguro quando um túnel é criado. O CAT é usado pelos dispositivos de origem e de destino para se conectar ao serviço de encapsulamento seguro.

Token de cliente

Um valor exclusivo gerado pelo cliente queAWS IoTo túnel seguro pode ser usado para todas as conexões de repetição subsequentes para o mesmo túnel. Esse campo é opcional. Se o token do cliente não for fornecido, o token de acesso do cliente (CAT) só poderá ser usado uma vez para o

mesmo túnel. Tentativas de conexão subsequentes usando o mesmo CAT serão rejeitadas. Para obter mais informações sobre como usar tokens do cliente, consulte oimplementação de referência de proxy local no GitHub.

Aplicativo de destino

O aplicativo que é executado no dispositivo de destino. Por exemplo, o aplicativo de destino pode ser um daemon SSH para estabelecer uma sessão SSH usando o encapsulamento seguro.

Dispositivo de destino

O dispositivo remoto que você deseja acessar.

Agente do dispositivo

Um aplicativo da IoT que se conecta ao gateway de dispositivos da AWS IoT e escuta novas notificações de encapsulamento pelo MQTT. Para obter mais informações, consulteSnippet de agente de IoT (p. 795)

Proxy local

Um proxy de software que é executado nos dispositivos de origem e de destino e retransmite um fluxo de dados entre encapsulamento seguro e o aplicativo do dispositivo. O proxy local pode ser executado no modo de origem ou no modo de destino. Para obter mais informações, consulteProxy local (p. 782)

Dispositivo de origem

O dispositivo que um operador usa para iniciar uma sessão no dispositivo de destino, geralmente um notebook ou computador desktop.

Túnel

Um caminho lógico pela AWS IoT que permite a comunicação bidirecional entre um dispositivo de origem e um dispositivo de destino.

Como o encapsulamento seguro do funciona

A seguir, mostra como o tunelamento seguro estabelece uma conexão entre o dispositivo de origem e de destino. Para obter informações sobre os diferentes termos, como token de acesso do cliente (CAT), consulteConceitos de encapsulamento seguro (p. 777).

1. Abrir um túnel

   Para abrir um túnel para iniciar uma sessão com seu dispositivo de destino remoto, você pode usar oAWS Management Console, oAWS CLItúnel abertoComando ou oAPI OpenTunnel.

2. Baixe o par de tokens de acesso para cliente

   Depois de abrir um túnel, você pode baixar o token de acesso do cliente (CAT) para sua origem e destino e salvá-lo em seu dispositivo de origem. Você deve recuperar o CAT e salvá-lo agora antes de iniciar o proxy local.

3. Iniciar proxy local no modo de destino

   O agente de IoT que foi instalado e está em execução no dispositivo de destino será inscrito no tópico MQTT reservado`$aws/things/`*`thing-name`*`/tunnels/notify`e receberá o CAT. Aqui,*`Nome da coisa`*-É o nome doAWS IoTcoisa que você cria para o seu destino. Para obter mais informações, consulteProteger tópicos de túneis (p. 114)

   Em seguida, o agente de IoT usa o CAT para iniciar o proxy local no modo de destino e configurar uma conexão no lado de destino do túnel. Para obter mais informações, consulteSnippet de agente de IoT (p. 795)

4. Iniciar proxy local no modo de origem

   Depois que o túnel for aberto,AWS IoT Device Managementfornece o CAT para a fonte que você pode baixar no dispositivo de origem. É possível usar o CAT para iniciar o proxy local no modo de origem, que então conecta o lado de origem do túnel. Para obter mais informações sobre proxy local, consulteProxy local (p. 782).
5. Abrir uma sessão SSH

   Como ambos os lados do túnel estão conectados, você pode iniciar uma sessão SSH usando o proxy local no lado de origem.

Para obter mais informações sobre como usar oAWS Management ConsolePara abrir um encapsulamento e iniciar uma sessão SSH, consulteAbra um túnel e inicie a sessão SSH no dispositivo remoto (p. 780).

O vídeo a seguir descreve como o tunelamento seguro funciona e o orienta pelo processo de configuração de uma sessão SSH para um dispositivo Raspberry Pi.

Ciclo de vida do túnel seguro

Os túneis podem ter o status`OPEN`ou`CLOSED`. Conexões com o túnel podem ter o status`CONNECTED`ou`DISCONNECTED`. A seguir, mostra como funcionam os diferentes status de túnel e conexão.

1. Ao abrir um túnel, ele terá o status `OPEN`. O status da conexão de origem e de destino do túnel é definido como `DISCONNECTED`.
2. Quando um dispositivo (origem ou de destino) se conecta ao túnel, o status da conexão correspondente muda para`CONNECTED`.
3. Quando um dispositivo se desconecta do túnel enquanto o status do túnel permanece`OPEN`, o status da conexão correspondente volta para`DISCONNECTED`. Um dispositivo pode se conectar e se desconectar de um túnel repetidamente enquanto o túnel permanece como `OPEN`.
4. Quando você liga`CloseTunnel`ou o túnel permanece`OPEN`por mais tempo que o`MaxLifetimeTimeout`valor, o status de um túnel se torna`CLOSED`. É possível configurar `MaxLifetimeTimeout` ao chamar `OpenTunnel`. `MaxLifetimeTimeout` assumirá o padrão de 12 horas se você não especificar um valor.

   > **Note**
   >
   > Um túnel não pode ser reaberto quando está no status `CLOSED`.
5. Você pode ligar`DescribeTunnel`e`ListTunnels`Para visualizar os metadados do túnel enquanto o túnel estiver visível. O túnel pode ser visível noAWS IoTPor pelo menos três horas antes de ser excluído.

AWS IoTTutoriais de encapsulamento seguro

AWS IoTO encapsulamento seguro ajuda os clientes a estabelecer comunicação bidirecional com dispositivos remotos que estão por trás do firewall por meio de uma conexão segura gerenciada pelaAWS IoT. Os tutoriais a seguir vão ajudá-lo a aprender como começar e usar encapsulamento seguro.

Para demonstração rápida doAWS IoTtunelamento seguro, use nossoAWS IoTdemonstração de tunelamento seguro no GitHub.

AWS IoTTutoriais de encapsulamento seguro

- Abra um túnel e inicie a sessão SSH no dispositivo remoto (p. 780)

Abra um túnel e inicie a sessão SSH no dispositivo remoto

Neste tutorial, abra um túnel e use-o para iniciar uma sessão SSH em um dispositivo remoto. O dispositivo remoto está atrás de firewalls que bloqueiam todo o tráfego de entrada, tornando impossível fazer o SSH direto para o dispositivo. Antes de começar, entenda como registar um dispositivo no registro da AWS IoT e conectar um dispositivo ao gateway de dispositivos da AWS IoT.

Pré-requisitos

- Os firewalls que protegem o dispositivo remoto devem permitir o tráfego de saída na porta 443.
- Você criou uma coisa da IoT chamada `RemoteDeviceA` no registro da AWS IoT.
- Você tem um agente de dispositivo da IoT em execução no dispositivo remoto que se conecta ao gateway de dispositivos da AWS IoT e está configurado com uma assinatura de tópico MQTT. Este tutorial inclui um trecho que mostra como implementar um agente. Para obter mais informações, consulte Snippet de agente de IoT (p. 795).
- É necessário ter um daemon SSH em execução no dispositivo remoto.
- Você fez download do código-fonte do proxy local do GitHub e o criou para a plataforma de sua escolha. Neste tutorial, vamos nos referir ao arquivo executável de proxy local que você criou como `localproxy`.

Abrir um túnel

Se você configurar o destino ao chamar`OpenTunnel`O encapsulamento seguro entregará o token de acesso de cliente de destino ao dispositivo remoto por MQTT e pelo tópico MQTT reservado (`$aws/things/RemoteDeviceA/tunnels/notify`). Para obter mais informações, consulte Tópicos reservados (p. 101). Após o recebimento da mensagem MQTT, o agente de IoT no dispositivo remoto inicia o proxy local no modo de destino. É possível omitir a configuração de destino se quiser entregar o token de acesso do cliente de destino ao dispositivo remoto por meio de outro método. Para obter mais informações, consulte Configurar um dispositivo remoto (p. 801).

Para abrir um túnel no console

1. NoAWS IoTconsole, navegue atéGerenciee escolhaTúneis.

2. SelecioneCriar túnel.
3. Insira uma descrição de túnel, o nome da coisa para o qual você deseja abrir um túnel, o serviço a ser usado, como SSH ou FTP, uma duração de tempo limite de túnel e tags de recurso como pares de chave-valor para ajudá-lo a identificar seu recurso.
4. Faça o download dos tokens de acesso do cliente para a origem e o destino. Os tokens não estarão disponíveis para download depois que você escolherConcluído.

5. Selecione Done (Concluído).

Iniciar o proxy local

Abra um terminal no notebook, copie o token de acesso do cliente de origem e use-o para iniciar o proxy local no modo de origem. No comando a seguir, o proxy local está configurado para escutar novas conexões na porta 5555.

./localproxy -r us-east-1 -s 5555 -t ***source-client-access-token***

> **Note**
>
> ORegião da AWSNeste comando deve ser o mesmoRegião da AWSOnde o túnel foi criado.

-r

Especifica oRegião da AWSOnde o túnel é criado.

-s

Especifica a porta à qual o proxy deve se conectar.

-t

Especifica o texto do token do cliente.

> **Note**
>
> Se você receber o erro a seguir, configure o caminho da CA. Para mais informações, consulte o GitHub.
>
> `Could not perform SSL handshake with proxy server: certificate verify failed`

Iniciar uma sessão SSH

Abra outro terminal e use o comando a seguir para iniciar uma nova sessão SSH conectando-se ao proxy local na porta 5555.

ssh ***username***@localhost -p 5555

Talvez seja solicitada uma senha para a sessão SSH. Quando finalizar a sessão SSH, digite **`exit`** para fechar a sessão.

Fechar o encapsulamento

1. Abra o console do AWS IoT.
2. Escolha o túnel e, em Actions (Ações), selecione Close (Fechar). Fechar o túnel faz com que ambas as instâncias de proxy locais fechem.

Proxy local

O proxy local transmite dados enviados pelo aplicativo em execução no dispositivo de origem usando tunelamento seguro em um WebSocket Conexão segura. É possível fazer download da origem de proxy local no GitHub.

O proxy local pode ser executado em dois modos: `source` ou `destination`. No modo de origem, o proxy local é executado no mesmo dispositivo ou na mesma rede do aplicativo cliente que inicia a conexão TCP. No modo de destino, o proxy local é executado no dispositivo remoto com o aplicativo de destino. Um único túnel pode suportar até três conexões TCP por vez usando a multiplexação de túnel. Para obter mais informações, consulteMultiplex fluxos de dados em um túnel seguro (p. 791)

Como usar o proxy local

Você pode executar o proxy local nos dispositivos de origem e de destino para transmitir dados para os endpoints de tunelamento seguro. Se seus dispositivos estiverem em uma rede que usa um proxy da Web, o proxy da Web poderá interceptar as conexões antes de encaminhá-las para a Internet. Nesse caso, você precisará configurar seu proxy local para usar o proxy da Web. Para obter mais informações, consulteConfigurar proxy local para dispositivos que usam proxy da web (p. 786)

Fluxo de trabalho proxy local

As etapas a seguir mostram como o proxy local é executado nos dispositivos de origem e de destino.

1. Connect o proxy local ao tunelamento seguro

   Primeiro, o proxy local deve estabelecer uma conexão para o túnel seguro. Ao iniciar o proxy local, use os seguintes argumentos:

   - O`-r`argumento para especificar oRegião da AWSno qual o túnel é aberto.
   - O`-t`Para transmitir o token de acesso do cliente de origem ou de destino retornado do`openTunnel`.

     > **Note**
     >
     > Dois proxies locais que usam o mesmo valor de token de acesso do cliente não podem ser conectados ao mesmo tempo.

2. Execute ações de origem ou de destino

   Depois do WebSocket A conexão é estabelecida, o proxy local executa ações do modo de origem ou de destino, dependendo da configuração.

   Por padrão, o proxy local tentará se reconectar para encapsulamento seguro se ocorrerem erros de entrada/saída (E/S) ou se o WebSocket a conexão é fechada inesperadamente. Isso faz com que a conexão TCP seja fechada. Se ocorrerem erros de soquete TCP, o proxy local enviará uma

mensagem pelo túnel a fim de notificar o outro lado para fechar sua conexão TCP. Por padrão, o proxy local sempre usa comunicação SSL.

3. Interromper o proxy local

   Depois de usar o túnel, é seguro interromper o processo de proxy local. Recomendamos que você feche explicitamente o túnel chamando `CloseTunnel`. Os clientes de túnel ativos podem não ser fechados imediatamente após a chamada `CloseTunnel`.

Para obter mais informações sobre como usar oAWS Management ConsolePara abrir um encapsulamento e iniciar uma sessão SSH, consulteAbra um túnel e inicie a sessão SSH no dispositivo remoto (p. 780).

Práticas recomendadas para proxy local

Ao executar o proxy local, siga estas melhores práticas:

- Evite o uso do argumento de proxy local `-t` para transmitir em um token de acesso. Recomendamos que você use a variável de ambiente `AWSIOT_TUNNEL_ACCESS_TOKEN` para definir o token de acesso para o proxy local.
- Execute o executável de proxy local com menos privilégios no sistema operacional ou no ambiente.
  - Evite executar o proxy local como administrador no Windows.
  - Evite executar o proxy local como raiz no Linux e no macOS.
- Considere executar o proxy local em hosts, contêineres, sandboxes, chroot jail separados ou em um ambiente virtualizado.
- Crie o proxy local com sinalizadores de segurança relevantes, dependendo da cadeia de ferramentas.
- Em dispositivos com várias interfaces de rede, use o argumento `-b` para vincular o soquete TCP à interface de rede usada para se comunicar com o aplicativo de destino.

Exemplo de comando e saída

A seguir, mostra um exemplo de um comando que você executa em um sistema operacional Linux e na saída correspondente. O exemplo mostra um proxy da Web que está escutando em uma porta HTTP e como o proxy local pode ser configurado em ambos`source`e`destination`Modos. Antes de executar esses comandos, você já deve ter aberto um túnel e obtido os tokens de acesso do cliente para a origem e o destino. Você também deve ter criado o proxy local, conforme descrito anteriormente.

O proxy local atualiza o protocolo HTTPS para WebSockets para estabelecer uma conexão de longa duração e, em seguida, começa a transmitir dados através da conexão aos pontos de extremidade do dispositivo de tunelamento seguro.

> **Note**
>
> Os comandos a seguir usados nos exemplos usam `overbosity`sinalizador para ilustrar uma visão geral das diferentes etapas descritas anteriormente após a execução do proxy local. Recomendamos usar esse sinalizador somente para fins de teste.

Executando proxy local no modo de origem

Os comandos a seguir mostram como executar o proxy local no modo de origem.

Linux/macOS

No Linux ou no macOS, execute os seguintes comandos no terminal para configurar e iniciar o proxy local na sua origem.

```
export AWSIOT_TUNNEL_ACCESS_TOKEN=${access_token}
```

```
./localproxy -s 5555 -v 5 -r us-west-2
```

Onde:

- `-s` é a porta de escuta de origem, que inicia o proxy local no modo de origem.
- `-v` é a verbosidade da saída, que pode ser um valor entre zero e seis.
- `-r` é a região do endpoint onde o túnel é aberto.

Para obter mais informações sobre os parâmetros, consulte Opções definidas usando argumentos de linha de comando.

Windows

No Windows, você configura o proxy local semelhante ao que você faz para Linux ou macOS, mas como você define as variáveis de ambiente é diferente das outras plataformas. Execute os seguintes comandos no `cmd` para configurar e iniciar o proxy local na sua origem.

```
set AWSIOT_TUNNEL_ACCESS_TOKEN=${access_token}
.\localproxy -s 5555 -v 5 -r us-west-2
```

Onde:

- `-s` é a porta de escuta de origem, que inicia o proxy local no modo de origem.
- `-v` é a verbosidade da saída, que pode ser um valor entre zero e seis.
- `-r` é a região do endpoint onde o túnel é aberto.

Para obter mais informações sobre os parâmetros, consulte Opções definidas usando argumentos de linha de comando.

A seguir, um exemplo de saída da execução do proxy local no `source` Modo.

```
...
...

Starting proxy in source mode
Attempting to establish web socket connection with endpoint wss://data.tunneling.iot.us-
west-2.amazonaws.com:443
Resolved proxy  server IP: 10.10.0.11
Connected successfully with proxy server
Performing SSL handshake with proxy server
Successfully completed SSL handshake with proxy server
HTTP/1.1 101 Switching Protocols

...

Connection: upgrade
channel-id: 01234567890abc23-00001234-0005678a-b1234c5de677a001-2bc3d456
upgrade: websocket

...

Web socket session ID: 01234567890abc23-00001234-0005678a-b1234c5de677a001-2bc3d456
Web socket subprotocol selected: aws.iot.securetunneling-2.0
Successfully established websocket connection with proxy server: wss://
data.tunneling.iot.us-west-2.amazonaws.com:443
Setting up web socket pings for every 5000 milliseconds
Scheduled next read:
```

```
...

Starting web socket read loop continue reading...
Resolved bind IP: 127.0.0.1
Listening for new connection on port 5555
```

Executando proxy local no modo de destino

Os comandos a seguir mostram como executar o proxy local no modo de destino.

Linux/macOS

No Linux ou no macOS, execute os seguintes comandos no terminal para configurar e iniciar o proxy local no seu destino.

```
export AWSIOT_TUNNEL_ACCESS_TOKEN=${access_token}
./localproxy -d 22 -v 5 -r us-west-2
```

Onde:

- `-d`É o aplicativo de destino que inicia o proxy local no modo de destino.
- `-v`é a verbosidade da saída, que pode ser um valor entre zero e seis.
- `-r`é a região do endpoint onde o túnel é aberto.

Para obter mais informações sobre os parâmetros, consulteOpções definidas usando argumentos de linha de comando.

Windows

No Windows, você configura o proxy local semelhante ao que você faz para Linux ou macOS, mas como você define as variáveis de ambiente é diferente das outras plataformas. Execute os seguintes comandos no`cmd`para configurar e iniciar o proxy local no seu destino.

```
set AWSIOT_TUNNEL_ACCESS_TOKEN=${access_token}
.\localproxy -d 22 -v 5 -r us-west-2
```

Onde:

- `-d`É o aplicativo de destino que inicia o proxy local no modo de destino.
- `-v`é a verbosidade da saída, que pode ser um valor entre zero e seis.
- `-r`é a região do endpoint onde o túnel é aberto.

Para obter mais informações sobre os parâmetros, consulteOpções definidas usando argumentos de linha de comando.

A seguir, um exemplo de saída da execução do proxy local no`destination`Modo.

```
...
...

Starting proxy in destination mode
Attempting to establish web socket connection with endpoint wss://data.tunneling.iot.us-
west-2.amazonaws.com:443
Resolved proxy  server IP: 10.10.0.11
Connected successfully with proxy server
Performing SSL handshake with proxy server
Successfully completed SSL handshake with proxy server
```

```
HTTP/1.1 101 Switching Protocols

...

Connection: upgrade
channel-id: 01234567890abc23-00001234-0005678a-b1234c5de677a001-2bc3d456
upgrade: websocket

...

Web socket session ID: 01234567890abc23-00001234-0005678a-b1234c5de677a001-2bc3d456
Web socket subprotocol selected: aws.iot.securetunneling-2.0
Successfully established websocket connection with proxy server: wss://
data.tunneling.iot.us-west-2.amazonaws.com:443
Setting up web socket pings for every 5000 milliseconds
Scheduled next read:

...

Starting web socket read loop continue reading...
```

Configurar proxy local para dispositivos que usam proxy da web

Você pode usar proxy local emAWS IoTdispositivos para se comunicar comAWS IoTAPIs de encapsulamento seguro. O proxy local transmite dados enviados pelo aplicativo do dispositivo usando encapsulamento seguro por um WebSocket Conexão segura. O proxy local pode funcionar em`source`ou`destination`Modo. Dentro`source`Em, ele é executado no mesmo dispositivo ou na mesma rede que inicia a conexão TCP. Dentro`destination`Em, o proxy local é executado no dispositivo remoto com o aplicativo de destino. Para obter mais informações, consulteProxy local (p. 782)

O proxy local precisa se conectar diretamente à internet para usarAWS IoTEncapsulamento seguro do. Para uma conexão TCP de longa duração com tunelamento seguro, o proxy local atualiza a solicitação HTTPS para estabelecer um WebSockets conexão com um dosterminais de conexão de dispositivo de tunelamento seguro.

Se seus dispositivos estiverem em uma redeque usa um proxy da web, o proxy da web pode interceptar as conexões antes de encaminhá-las para a internet. Para estabelecer uma conexão de longa duração com os pontos de extremidade de conexão de dispositivo de tunelamento seguro, configure seu proxy local para usar o proxy da Web, conforme descrito naEspecificação websocket.

> **Note**
>
> OAWS IoTCliente de dispositivo (p. 1281)Não oferece suporte a dispositivos que usam um proxy web. Para trabalhar com o proxy da Web, você precisará usar um proxy local e configurá-lo para trabalhar com um proxy da Web, conforme descrito abaixo.

As etapas a seguir mostram como o proxy local funciona com um proxy da Web.

1. O proxy local envia um HTTP`CONNECT`solicitação para o proxy da Web que contém o endereço remoto do serviço de tunelamento seguro, juntamente com as informações de autenticação do proxy da Web.
2. O proxy da Web criará uma conexão de longa duração com os pontos de extremidade de tunelamento seguro remoto.
3. A conexão TCP é estabelecida e o proxy local agora funcionará nos modos de origem e destino para transmissão de dados.

Para concluir esse procedimento, execute as seguintes etapas.

- Construa o proxy local (p. 787)

- Configurar seu proxy da web (p. 787)
- Configure e inicie o proxy local (p. 788)

Construa o proxy local

Abrir ocódigo-fonte proxy localno GitHub repositório e siga as instruções para criar e instalar o proxy local.

Configurar seu proxy da web

O proxy local depende do mecanismo de tunelamento HTTP descrito peloEspecificação HTTP/1.1. Para cumprir as especificações, seu proxy da Web deve permitir que os dispositivos usem o`CONNECT`Method.

A forma como você configura seu proxy da Web depende do proxy da Web que você está usando e da versão do proxy da Web. Para garantir que você configure o proxy da Web corretamente, verifique a documentação do proxy da Web.

Para configurar seu proxy da Web, primeiro identifique o URL do proxy da Web e confirme se o proxy da Web é compatível com tunelamento HTTP. O URL do proxy da Web será usado posteriormente quando você configurar e iniciar o proxy local.

1. Identifique o URL do proxy da Web

   O URL do proxy da web estará no formato a seguir.

   ```
   protocol://web_proxy_host_domain:web_proxy_port
   ```

   AWS IoTo tunelamento seguro é compatível apenas com autenticação básica para proxy da Web. Para usar a autenticação básica, você deve especificar o**`username`**e**`password`**como parte do URL do proxy da web. O URL do proxy da web estará no formato a seguir.

   ```
   protocol://username:password@web_proxy_host_domain:web_proxy_port
   ```

   - *`protocolo`*Pode ser`http`ou`https`. Recomendamos usar o `https`.
   - *`web_proxy_host_domain`*É o endereço IP do proxy da Web ou de um nome DNS que seja resolvido para o endereço IP do proxy da Web.
   - *`web_proxy_port`*É a porta na qual o proxy da web está ouvindo.
   - O proxy da web usa isso**`username`**e**`password`**Para autenticar a solicitação.
2. Teste o URL do proxy da web

   Para confirmar se o proxy da Web é compatível com tunelamento TCP, use um`curl`Comando e verifique se você recebe um`2xx`ou um`3xx`Resposta.

   Por exemplo, se o URL do proxy da Web for`https://server.com:1235`, use um`proxy-insecure`Sinalizador com o`curl`Comando porque o proxy da Web pode confiar em um certificado autoassinado.

   ```
   export HTTPS_PROXY=https://server.com:1235
   curl -I https://aws.amazon.com --proxy-insecure
   ```

   Se o URL do proxy da Web tiver um`http`port (por exemplo,`http://server.com:1234`), não é necessário usar o`proxy-insecure`Sinalizador.

   ```
   export HTTPS_PROXY=http://server.com:1234
   curl -I https://aws.amazon.com
   ```

Configure e inicie o proxy local

Para configurar o proxy local para usar um proxy da Web, você deve configurar o`HTTPS_PROXY`Variável de ambiente com os nomes de domínio DNS ou endereços IP e números de porta usados pelo proxy da Web.

Depois de configurar o proxy local, você pode usar o proxy local conforme explicado nesteLEIAMEdocument.

> **Note**
>
> Sua declaração de variável de ambiente diferencia maiúsculas. Recomendamos que você defina cada variável uma vez usando todas as letras maiúsculas ou minúsculas. Os exemplos a seguir mostram a variável de ambiente declarada em letras maiúsculas. Se a mesma variável for especificada usando letras maiúsculas e minúsculas, a variável especificada usando letras minúsculas terá precedência.

Os comandos a seguir mostram como configurar o proxy local que está sendo executado no seu destino para usar o proxy da Web e iniciar o proxy local.

- `AWSIOT_TUNNEL_ACCESS_TOKEN`: Essa variável contém o token de acesso do cliente (CAT) para o destino.
- `HTTPS_PROXY`: Essa variável contém o URL do proxy da Web ou o endereço IP para configurar o proxy local.

Os comandos mostrados nos exemplos a seguir dependem do sistema operacional que você usa e se o proxy da Web está escutando em uma porta HTTP ou HTTPS.

Proxy da Web escutando em uma porta HTTP

Se o proxy da Web estiver escutando em uma porta HTTP, você poderá fornecer o URL do proxy da Web ou o endereço IP para o`HTTPS_PROXY`Variável.

Linux/macOS

No Linux ou no macOS, execute os seguintes comandos no terminal para configurar e iniciar o proxy local no seu destino para usar um proxy da Web ouvindo uma porta HTTP.

```
export AWSIOT_TUNNEL_ACCESS_TOKEN=${access_token}
export HTTPS_PROXY=http:proxy.example.com:1234
./localproxy -r us-east-1 -d 22
```

Se você tiver que se autenticar com o proxy, você deve especificar um**username**e**password**Como parte do`HTTPS_PROXY`Variável.

```
export AWSIOT_TUNNEL_ACCESS_TOKEN=${access_token}
export HTTPS_PROXY=http://username:password@proxy.example.com:1234
./localproxy -r us-east-1 -d 22
```

Windows

No Windows, você configura o proxy local semelhante ao que você faz para Linux ou macOS, mas como você define as variáveis de ambiente é diferente das outras plataformas. Execute os seguintes comandos no`cmd`para configurar e iniciar o proxy local no seu destino para usar um proxy da Web ouvindo uma porta HTTP.

```
set AWSIOT_TUNNEL_ACCESS_TOKEN=${access_token}
set HTTPS_PROXY=http://proxy.example.com:1234
```

```
.\localproxy -r us-east-1 -d 22
```

Se você tiver que se autenticar com o proxy, você deve especificar um**username**e**password**Como parte do`HTTPS_PROXY`Variável.

```
set AWSIOT_TUNNEL_ACCESS_TOKEN=${access_token}
set HTTPS_PROXY=http://username:password@10.15.20.25:1234
.\localproxy -r us-east-1 -d 22
```

Proxy da Web escutando em uma porta HTTPS

Execute os seguintes comandos se o proxy da Web estiver escutando em uma porta HTTPS.

> **Note**
>
> Se você estiver usando um certificado autoassinado para o proxy da Web ou se estiver executando o proxy local em um sistema operacional que não tenha suporte nativo do OpenSSL e configurações padrão, você terá que configurar seus certificados de proxy da Web conforme descrito naConfiguração do certificadoSeção no GitHub repositório do.

Os comandos a seguir serão semelhantes a como você configurou seu proxy da Web para um proxy HTTP, com a exceção de que você também especificará o caminho para os arquivos de certificado que você instalou conforme descrito anteriormente.

Linux/macOS

No Linux ou no macOS, execute os seguintes comandos no terminal para configurar o proxy local em execução no seu destino para usar um proxy da Web ouvindo uma porta HTTPS.

```
export AWSIOT_TUNNEL_ACCESS_TOKEN=${access_token}
export HTTPS_PROXY=http:proxy.example.com:1234
./localproxy -r us-east-1 -d 22 -c /path/to/certs
```

Se você tiver que se autenticar com o proxy, você deve especificar um**username**e**password**Como parte do`HTTPS_PROXY`Variável.

```
export AWSIOT_TUNNEL_ACCESS_TOKEN=${access_token}
export HTTPS_PROXY=http://username:password@proxy.example.com:1234
./localproxy -r us-east-1 -d 22 -c /path/to/certs
```

Windows

No Windows, execute os seguintes comandos no`cmd`para configurar e iniciar o proxy local em execução no seu destino para usar um proxy da Web ouvindo uma porta HTTP.

```
set AWSIOT_TUNNEL_ACCESS_TOKEN=${access_token}
set HTTPS_PROXY=http://proxy.example.com:1234
.\localproxy -r us-east-1 -d 22 -c \path\to\certs
```

Se você tiver que se autenticar com o proxy, você deve especificar um**username**e**password**Como parte do`HTTPS_PROXY`Variável.

```
set AWSIOT_TUNNEL_ACCESS_TOKEN=${access_token}
set HTTPS_PROXY=http://username:password@10.15.20.25:1234
.\localproxy -r us-east-1 -d 22 -c \path\to\certs
```

Exemplo de comando e saída

A seguir, mostra um exemplo de um comando que você executa em um sistema operacional Linux e na saída correspondente. O exemplo mostra um proxy da Web que está escutando em uma porta HTTP e como o proxy local pode ser configurado para usar o proxy da Web em ambos`sourceedestination`Modos. Antes de executar esses comandos, você já deve ter aberto um túnel e obtido os tokens de acesso do cliente para a origem e o destino. Você também deve ter criado o proxy local e configurado seu proxy da Web conforme descrito anteriormente.

Aqui está uma visão geral das etapas após iniciar o proxy local. O proxy local:

- Identifica o URL do proxy da Web para que ele possa usar a URL para se conectar ao servidor proxy.
- Estabelece uma conexão TCP com o proxy da web.
- Envia um HTTP`CONNECT`solicitação para o proxy da Web e aguarda o`HTTP/1.1 200`resposta, que indica que a conexão foi estabelecida.
- Atualiza o protocolo HTTPS para WebSockets Para estabelecer uma conexão de longa duração.
- Começa a transmitir dados através da conexão aos pontos de extremidade do dispositivo de tunelamento seguro.

> **Note**
>
> Os comandos a seguir usados nos exemplos usam `overbosity`sinalizador para ilustrar uma visão geral das diferentes etapas descritas anteriormente após a execução do proxy local. Recomendamos usar esse sinalizador somente para fins de teste.

Executando proxy local no modo de origem

Os comandos a seguir mostram como executar o proxy local no modo de origem.

```
export AWSIOT_TUNNEL_ACCESS_TOKEN=${access_token}
export HTTPS_PROXY=http:username:password@10.15.10.25:1234
./localproxy -s 5555 -v 5 -r us-west-2
```

A seguir, um exemplo de saída da execução do proxy local no`source`Modo.

```
...

Parsed basic auth credentials for the URL
Found Web proxy information in the environment variables, will use it to connect via the
 proxy.

...

Starting proxy in source mode
Attempting to establish web socket connection with endpoint wss://data.tunneling.iot.us-
west-2.amazonaws.com:443
Resolved Web proxy IP: 10.10.0.11
Connected successfully with Web Proxy
Successfully sent HTTP CONNECT to the Web proxy
Full response from the Web proxy:
HTTP/1.1 200 Connection established
TCP tunnel established successfully
Connected successfully with proxy server
Successfully completed SSL handshake with proxy server
Web socket session ID: 0a109afffee745f5-00001341-000b8138-cc6c878d80e8adb0-f186064b
Web socket subprotocol selected: aws.iot.securetunneling-2.0
Successfully established websocket connection with proxy server: wss://
data.tunneling.iot.us-west-2.amazonaws.com:443
Seting up web socket pings for every 5000 milliseconds
Scheduled next read:
```

```
...

Starting web socket read loop continue reading...
Resolved bind IP: 127.0.0.1
Listening for new connection on port 5555
```

Executando proxy local no modo de destino

Os comandos a seguir mostram como executar o proxy local no modo de destino.

```
export AWSIOT_TUNNEL_ACCESS_TOKEN=${access_token}
export HTTPS_PROXY=http:username:password@10.15.10.25:1234
./localproxy -d 22 -v 5 -r us-west-2
```

A seguir, um exemplo de saída da execução do proxy local no`destination`Modo.

```
...

Parsed basic auth credentials for the URL
Found Web proxy information in the environment variables, will use it to connect via the
 proxy.

...

Starting proxy in destination mode
Attempting to establish web socket connection with endpoint wss://data.tunneling.iot.us-
west-2.amazonaws.com:443
Resolved Web proxy IP: 10.10.0.1
Connected successfully with Web Proxy
Successfully sent HTTP CONNECT to the Web proxy
Full response from the Web proxy:
HTTP/1.1 200 Connection established
TCP tunnel established successfully
Connected successfully with proxy server
Successfully completed SSL handshake with proxy server
Web socket session ID: 06717bfffed3fd05-00001355-000b8315-da3109a85da804dd-24c3d10d
Web socket subprotocol selected: aws.iot.securetunneling-2.0
Successfully established websocket connection with proxy server: wss://
data.tunneling.iot.us-west-2.amazonaws.com:443
Seting up web socket pings for every 5000 milliseconds
Scheduled next read:

...

Starting web socket read loop continue reading...
```

Multiplex fluxos de dados em um túnel seguro

Você pode usar vários fluxos de dados por túnel usando o recurso de multiplexação de túnel seguro. Com a multiplexação, você pode solucionar problemas de dispositivos usando várias conexões ou portas (por exemplo, um navegador da Web que requer o envio de vários fluxos de dados HTTP e SSH). Você também pode reduzir sua carga operacional eliminando a necessidade de criar, implantar e iniciar vários proxies locais ou abrir vários túneis para o mesmo dispositivo.

Exemplo de caso de uso

Você pode usar o recurso de multiplexação caso um dispositivo no campo exija mais de uma conexão com o dispositivo para solucioná-lo corretamente. Por exemplo, talvez seja necessário se conectar a um aplicativo da Web no dispositivo para alterar alguns parâmetros de rede, enquanto simultaneamente emite

comandos shell através do terminal para verificar se o dispositivo está funcionando corretamente com os novos parâmetros de rede. Nesse cenário, talvez seja necessário se conectar ao dispositivo por meio de HTTP e SSH e transferir dois fluxos de dados parallel para acessar simultaneamente o aplicativo da Web e o terminal. Com o recurso de multiplexação, esses dois fluxos independentes podem ser transferidos pelo mesmo túnel ao mesmo tempo.

Como configurar um túnel multiplexado

O procedimento a seguir orientará você sobre como configurar um túnel multiplexado para solucionar problemas de dispositivos usando aplicativos que exigem conexões com várias portas. Você configurará um túnel com dois fluxos multiplexados: um fluxo HTTP e um fluxo SSH.

1. Primeiro, configure o dispositivo de destino com arquivos de configuração. Os arquivos de configuração podem ser fornecidos no dispositivo se os mapeamentos de porta não forem alterados. No dispositivo de destino, crie um diretório de configuração chamado`config`na mesma pasta em que o proxy local está sendo executado. Em seguida, crie um arquivo chamado`SSHSource.ini`Neste diretório. O conteúdo desse arquivo é:

   ```
   HTTP1 = 5555
   SSH1 = 3333
   ```

   > **Note**
   >
   > Você pode pular esta etapa se preferir especificar o mapeamento de porta por meio da CLI ou não precisar iniciar o proxy local em portas de escuta designadas.

2. Em seguida, configure o dispositivo de origem com arquivos de configuração. Na mesma pasta em que o proxy local está sendo executado, crie um diretório de configuração chamado`config`e dê ao proxy local a permissão de leitura para este diretório. Em seguida, crie um arquivo chamado`SSHDestination.ini`Neste diretório. O conteúdo desse arquivo é:

   ```
   HTTP1 = 80
   SSH1 = 22
   ```

   > **Note**
   >
   > É possível ignorar esta etapa se preferir especificar o mapeamento da porta por meio da CLI. Em caso afirmativo, você precisará atualizar oAgente de túnel (p. 795)Para usar os novos parâmetros.

3. Abra um túnel com o identificador de serviço`HTTP1`e`SSH1`.`thingName`é opcional se seu dispositivo não estiver registrado comAWS IoT.

   ```
   aws iotsecuretunneling open-tunnel \
   --destination-config thingName=foo,services=HTTP1,SSH1
   ```

   Um token de acesso para cliente de origem e destino será fornecido após essa chamada. Observe o`destination_client_access_token`e`source_client_access_token`para as próximas etapas. A saída deve ser semelhante à seguinte.

   ```
   {
    "tunnelId": "b2de92a3-b8ff-46c0-b0f2-afa28b00cecd",
    "tunnelArn": "arn:aws:iot:us-west-2:431600097591:tunnel/b2de92a3-b8ff-46c0-b0f2-
   afa28b00cecd",
    "sourceAccessToken": source_client_access_token,
    "destinationAccessToken": destination_client_access_token
   }
   ```

4. Em seguida, inicie o proxy local de destino. Você estará conectado ao serviço de tunelamento seguro após a entrega do token. Um proxy local em execução em dispositivos de destino começa no modo de destino. Você tem duas opções para conseguir isso:

   1. Inicie o proxy local de destino com arquivos de configuração da Etapa 1.

   ```
   ./localproxy -r us-east-1 -m dst -t destination_client_access_token
   ```

   2. Inicie o proxy local de destino com o mapeamento especificado por meio da CLI.

   ```
   ./localproxy -r us-east-1 -d HTTP1=80,SSH1=22 -t destination_client_access_token
   ```

5. Agora, inicie o proxy local de origem. Um proxy local em execução em dispositivos de origem começa no modo de origem. Você tem três opções para conseguir isso:

   1. Inicie o proxy local de origem com arquivos de configuração da Etapa 2.

   ```
   ./localproxy -r us-east-1 -m src -t source_client_access_token
   ```

   2. Inicie o proxy local de origem com o mapeamento especificado por meio da CLI.

   ```
   ./localproxy -r us-east-1 -s HTTP1=5555,SSH1=3333 -t source_client_access_token
   ```

   3. Inicie o proxy local de origem sem arquivos de configuração e nenhum mapeamento especificado a partir da CLI. O proxy local retirará as portas disponíveis para usar e gerenciar os mapeamentos para você.

   ```
   ./localproxy -r us-east-1 -m src -t source_client_access_token
   ```

6. Os dados do aplicativo da conexão SSH e HTTP agora podem ser transferidos simultaneamente pelo túnel multiplexado. Como pode ser visto no mapa abaixo, o identificador de serviço atua como um formato legível para traduzir o mapeamento de porta entre o dispositivo de origem e de destino. Com essa configuração, o tunelamento seguro:

   1. Encaminhe qualquer tráfego HTTP de entrada da porta 5555 no dispositivo de origem para a porta 80 no dispositivo de destino.
   2. Encaminhe qualquer tráfego SSH de entrada da porta 3333 no dispositivo de origem para a porta 22 no dispositivo de destino.

Source Local Proxy
Service identifiers mapping:
HTTP1
5555
SSH1
3333
Option1
Option2
Option3
Configuration file
CLI
Local Proxy generates
Translation after processing:
5555
3333

Snippet de agente de IoT

O agente da IoT é usado para receber a mensagem MQTT que inclui o token de acesso do cliente e iniciar um proxy local no dispositivo remoto. É necessário instalar e executar o agente da IoT no dispositivo remoto se deseja que o encapsulamento seguro forneça o token de acesso do cliente. O agente da IoT deve assinar o seguinte tópico MQTT reservado da IoT:

```
$aws/things/thing-name/tunnels/notify
```

Em que `thing-name` é o nome da coisa da IoT associada ao dispositivo remoto.

Veja seguir um exemplo de carga da mensagem MQTT:

```
{
    "clientAccessToken": "destination-client-access-token",
    "clientMode": "destination",
    "region": "aws-region",
    "services": ["destination-service"]
}
```

Depois de receber uma mensagem MQTT, o agente da IoT deve iniciar um proxy local no dispositivo remoto com os parâmetros apropriados.

O seguinte código Java demonstra como usar oAWS IoTSDK de dispositivo daeProcessBuilderNa biblioteca Java para criar um agente da IoT simples para trabalhar com encapsulamento seguro.

```
// Find the IoT device endpoint for your Conta da AWS
final String endpoint = iotClient.describeEndpoint(new
 DescribeEndpointRequest().withEndpointType("iot:Data-ATS")).getEndpointAddress();

// Instantiate the IoT Agent with your AWS credentials
final String thingName = "RemoteDeviceA";
final String tunnelNotificationTopic = String.format("$aws/things/%s/tunnels/notify",
 thingName);
final AWSIotMqttClient mqttClient = new AWSIotMqttClient(endpoint, thingName,
                 "your_aws_access_key", "your_aws_secret_key");

try {
    mqttClient.connect();
    final TunnelNotificationListener listener = new
 TunnelNotificationListener(tunnelNotificationTopic);
    mqttClient.subscribe(listener, true);
}
finally {
    mqttClient.disconnect();
}

private static class TunnelNotificationListener extends AWSIotTopic {
    public TunnelNotificationListener(String topic) {
        super(topic);
    }

    @Override
    public void onMessage(AWSIotMessage message) {
        try {
            // Deserialize the MQTT message
            final JSONObject json = new JSONObject(message.getStringPayload());

            final String accessToken = json.getString("clientAccessToken");
            final String region = json.getString("region");
```

```
            final String clientMode = json.getString("clientMode");
            if (!clientMode.equals("destination")) {
                throw new RuntimeException("Client mode " + clientMode + " in the MQTT
 message is not expected");
            }

            final JSONArray servicesArray = json.getJSONArray("services");
            if (servicesArray.length() > 1) {
                throw new RuntimeException("Services in the MQTT message has more than 1
 service");
            }
            final String service = servicesArray.get(0).toString();
            if (!service.equals("SSH")) {
                throw new RuntimeException("Service " + service + " is not supported");
            }

            // Start the destination local proxy in a separate process to connect to the
 SSH Daemon listening port 22
            final ProcessBuilder pb = new ProcessBuilder("localproxy",
                        "-t", accessToken,
                        "-r", region,
                        "-d", "localhost:22");
            pb.start();
        }
        catch (Exception e) {
            log.error("Failed to start the local proxy", e);
        }
    }
}
```

Controle de acesso aos túneis

O encapsulamento seguro fornece ações, recursos e chaves de contexto de condição específicos do serviço para uso em políticas de permissão do IAM.

Pré-requisitos de acesso ao túnel

- Saiba como protegerAWSRecursos usando oPolíticas do IAM.
- Saiba como criar e avaliar as condições do IAM.
- Saiba como protegerAWSRecursos usando oTags de recursos.

Políticas de acesso ao túnel

Você deve usar as seguintes políticas para autorizar permissões para usar a API de tunelamento seguro. Para obter mais informações sobre a segurança da AWS IoT, consulte Identity and Access Management para o AWS IoT (p. 383).

iot:OpenTunnel

A ação de política `iot:OpenTunnel` concede permissão a um principal para chamar OpenTunnel.

No`Resource`elemento da declaração de política do IAM:

- Especifique o ARN do túnel curinga:

  `arn:aws:iot:`*`aws-region`*`:`*`aws-account-id`*`:tunnel/*`

- Especifique uma coisa ARN para gerenciar o`OpenTunnel`permissão para coisas específicas da IoT:

  `arn:aws:iot:`*`aws-region`*`:`*`aws-account-id`*`:thing/`*`thing-name`*

Por exemplo, a instrução da política a seguir permite abrir um túnel para a coisa da IoT chamada `TestDevice`.

```
{
    "Effect": "Allow",
    "Action": "iot:OpenTunnel",
    "Resource": [
        "arn:aws:iot:aws-region:aws-account-id:tunnel/*",
        "arn:aws:iot:aws-region:aws-account-id:thing/TestDevice"
    ]
}
```

A ação da política `iot:OpenTunnel` oferece suporte às seguintes chaves de condição:

- `iot:ThingGroupArn`
- `iot:TunnelDestinationService`
- `aws:RequestTag/`*`tag-key`*
- `aws:SecureTransport`
- `aws:TagKeys`

A instrução de política a seguir permite abrir um túnel para a coisa caso ela pertença a um grupo de coisas com um nome que comece com`TestGroup`e o serviço de destino configurado no túnel é SSH.

```
{
    "Effect": "Allow",
    "Action": "iot:OpenTunnel",
    "Resource": [
        "arn:aws:iot:aws-region:aws-account-id:tunnel/*"
    ],
    "Condition": {
        "ForAnyValue:StringLike": {
            "iot:ThingGroupArn": [
                "arn:aws:iot:aws-region:aws-account-id:thinggroup/TestGroup*"
            ]
        },
        "ForAllValues:StringEquals": {
            "iot:TunnelDestinationService": [
                "SSH"
            ]
        }
    }
}
```

Também é possível usar tags de recurso para controlar a permissão para abrir túneis. Por exemplo, a instrução de política a seguir permite que um túnel seja aberto caso a chave de tag `Owner` estiver presente com um valor de `Admin` e nenhuma outra tag for especificada. Para obter informações gerais sobre como usar tags, consulteMarcar recursos do AWS IoT (p. 288).

```
{
    "Effect": "Allow",
    "Action": "iot:OpenTunnel",
    "Resource": [
        "arn:aws:iot:aws-region:aws-account-id:tunnel/*"
    ],
```

```
    "Condition": {
        "StringEquals": {
            "aws:RequestTag/Owner": "Admin"
        },
        "ForAllValues:StringEquals": {
            "aws:TagKeys": "Owner"
        }
    }
}
```

IOT: RotatetunnelAccessToken

O`iot:RotateTunnelAccessToken`A ação de política concede permissão a um principal para chamarRotateTunnelAccessToken.

No`Resource`elemento da declaração de política do IAM:

- Especifique um ARN de túnel totalmente qualificado:

  `arn:aws:iot:`*`aws-region`*`:` *`aws-account-id`*`:tunnel/`*`tunnel-id`*

  Você também pode usar o encapsulamento curinga ARN:

  `arn:aws:iot:`*`aws-region`*`:`*`aws-account-id`*`:tunnel/*`
- Especifique uma coisa ARN para gerenciar o`RotateTunnelAccessToken`permissão para coisas específicas da IoT:

  `arn:aws:iot:`*`aws-region`*`:`*`aws-account-id`*`:thing/`*`thing-name`*

Por exemplo, a instrução de política a seguir permite girar o token de acesso de origem de um túnel ou o token de acesso de destino do cliente para a coisa da IoT chamada`TestDevice`.

```
{
    "Effect": "Allow",
    "Action": "iot:RotateTunnelAccessToken",
    "Resource": [
        "arn:aws:iot:aws-region:aws-account-id:tunnel/*",
        "arn:aws:iot:aws-region:aws-account-id:thing/TestDevice"
    ]
}
```

A ação da política `iot:RotateTunnelAccessToken` oferece suporte às seguintes chaves de condição:

- `iot:ThingGroupArn`
- `iot:TunnelDestinationService`
- `iot:ClientMode`
- `aws:SecureTransport`

A instrução de política a seguir permite girar o token de acesso de destino para a coisa caso ela pertença a um grupo de coisas com um nome que comece com`TestGroup`, o serviço de destino configurado no túnel é SSH e o cliente está em`DESTINATION`Modo.

```
{
    "Effect": "Allow",
    "Action": "iot:RotateTunnelAccessToken",
    "Resource": [
        "arn:aws:iot:aws-region:aws-account-id:tunnel/*"
```

```
    ],
    "Condition": {
        "ForAnyValue:StringLike": {
            "iot:ThingGroupArn": [
                "arn:aws:iot:aws-region:aws-account-id:thinggroup/TestGroup*"
            ]
        },
        "ForAllValues:StringEquals": {
            "iot:TunnelDestinationService": [
                "SSH"
            ],
            "iot:ClientMode": "DESTINATION"
        }
    }
}
```

iot:DescribeTunnel

A ação de política `iot:DescribeTunnel` concede permissão a um principal para chamar DescribeTunnel.

No`Resource`elemento da declaração de política do IAM, especifique um túnel de túnel totalmente qualificado ARN:

`arn:aws:iot:`*`aws-region`*`:` *`aws-account-id`*`:tunnel/`*`tunnel-id`*

Você também pode usar o encapsulamento curinga ARN:

`arn:aws:iot:`*`aws-region`*`:`*`aws-account-id`*`:tunnel/*`

A ação da política `iot:DescribeTunnel` oferece suporte às seguintes chaves de condição:

- `aws:ResourceTag/`*`tag-key`*
- `aws:SecureTransport`

A instrução de política a seguir permite que você chame `DescribeTunnel` caso o túnel solicitado esteja marcado com a chave `Owner` com um valor de `Admin`.

```
{
    "Effect": "Allow",
    "Action": "iot:DescribeTunnel",
    "Resource": [
        "arn:aws:iot:aws-region:aws-account-id:tunnel/*"
    ],
    "Condition": {
        "StringEquals": {
            "aws:ResourceTag/Owner": "Admin"
        }
    }
}
```

iot:ListTunnels

A ação de política `iot:ListTunnels` concede permissão a um principal para chamar ListTunnels.

No`Resource`elemento da declaração de política do IAM:

- Especifique o ARN do túnel curinga:

  `arn:aws:iot:`*`aws-region`*`:`*`aws-account-id`*`:tunnel/*`

- Especifique uma coisa ARN para gerenciar `oListTunnels`permissão em coisas de IoT selecionadas:

  `arn:aws:iot:`*`aws-region`*`:`*`aws-account-id`*`:thing/`*`thing-name`*

O`iot:ListTunnels`A ação da política oferece suporte à chave de condição`aws:SecureTransport`.

A instrução de política a seguir permite listar túneis para a coisa chamada `TestDevice`.

```
{
    "Effect": "Allow",
    "Action": "iot:ListTunnels",
    "Resource": [
        "arn:aws:iot:aws-region:aws-account-id:tunnel/*",
        "arn:aws:iot:aws-region:aws-account-id:thing/TestDevice"
    ]
}
```

iot:ListTagsForResource

A ação de política `iot:ListTagsForResource` concede permissão a um principal para chamar `ListTagsForResource`.

No`Resource`elemento da declaração de política do IAM, especifique um túnel de túnel totalmente qualificado ARN:

`arn:aws:iot:`*`aws-region`*`: `*`aws-account-id`*`:tunnel/`*`tunnel-id`*

Você também pode usar o encapsulamento curinga ARN:

`arn:aws:iot:`*`aws-region`*`:`*`aws-account-id`*`:tunnel/*`

O`iot:ListTagsForResource`A ação da política oferece suporte à chave de condição`aws:SecureTransport`.

iot:CloseTunnel

A ação de política `iot:CloseTunnel` concede permissão a um principal para chamar CloseTunnel.

No`Resource`elemento da declaração de política do IAM, especifique um túnel de túnel totalmente qualificado ARN:

`arn:aws:iot:`*`aws-region`*`: `*`aws-account-id`*`:tunnel/`*`tunnel-id`*

Você também pode usar o encapsulamento curinga ARN:

`arn:aws:iot:`*`aws-region`*`:`*`aws-account-id`*`:tunnel/*`

A ação da política `iot:CloseTunnel` oferece suporte às seguintes chaves de condição:

- `iot:Delete`
- `aws:ResourceTag/`*`tag-key`*
- `aws:SecureTransport`

A instrução de política a seguir permite que você chame `CloseTunnel` caso o parâmetro `Delete` da solicitação seja `false` e o túnel solicitado esteja marcado com a chave `Owner` com um valor de `QATeam`.

```
{
```

```
    "Effect": "Allow",
    "Action": "iot:CloseTunnel",
    "Resource": [
        "arn:aws:iot:aws-region:aws-account-id:tunnel/*"
    ],
    "Condition": {
        "Bool": {
            "iot:Delete": "false"
        },
        "StringEquals": {
            "aws:ResourceTag/Owner": "QATeam"
        }
    }
}
```

iot:TagResource

A ação de política `iot:TagResource` concede permissão a um principal para chamar `TagResource`.

No`Resource`elemento da declaração de política do IAM, especifique um túnel de túnel totalmente qualificado ARN:

`arn:aws:iot:`*`aws-region`*`:` *`aws-account-id`*`:tunnel/`*`tunnel-id`*

Você também pode usar o encapsulamento curinga ARN:

`arn:aws:iot:`*`aws-region`*`:`*`aws-account-id`*`:tunnel/*`

O`iot:TagResource`A ação da política oferece suporte à chave de condição`aws:SecureTransport`.

iot:UntagResource

A ação de política `iot:UntagResource` concede permissão a um principal para chamar `UntagResource`.

No`Resource`elemento da declaração de política do IAM, especifique um túnel de túnel totalmente qualificado ARN:

`arn:aws:iot:`*`aws-region`*`:` *`aws-account-id`*`:tunnel/`*`tunnel-id`*

Você também pode usar o encapsulamento curinga ARN:

`arn:aws:iot:`*`aws-region`*`:`*`aws-account-id`*`:tunnel/*`

O`iot:UntagResource`A ação da política oferece suporte à chave de condição`aws:SecureTransport`.

Configurar um dispositivo remoto

Se quiser entregar o token de acesso do cliente de destino ao dispositivo remoto por outros métodos diferentes da assinatura do tópico MQTT reservado da IoT, poderá ser necessário ter dois componentes no dispositivo remoto:

- Um listener de token de acesso do cliente de destino.
- Um proxy local.

O listener de token de acesso do cliente de destino deve funcionar com o mecanismo de entrega do token de acesso do cliente de sua escolha. Ele deve ser capaz de iniciar um proxy local no modo de destino.

ResolvedorAWS IoTproblemas de conectividade de tunelamento seguro girando tokens de acesso para cliente

Quando você usaAWS IoTtúnel seguro, você pode encontrar problemas de conectividade mesmo se o túnel estiver aberto. As seções a seguir mostram alguns possíveis problemas e como você pode resolvê-los girando os tokens de acesso do cliente. Para girar o token de acesso do cliente (CAT), use oRotateTunnelAccessTokenAPI ou o comandotoken de acesso de túnel rotativo AWS CLI. Dependendo se você se deparar com um erro ao usar o cliente no modo de origem ou de destino, você pode girar o CAT no modo de origem ou de destino, ou em ambos.

> **Note**
>
> - Se você não tiver certeza se o CAT precisa ser girado na origem ou no destino, você pode girar o CAT na origem e no destino definindo`ClientMode`para ALL ao usar `oRotateTunnelAccessToken`API.
> - Girar o CAT não estende a duração do túnel. Por exemplo, digamos que a duração do túnel seja de 12 horas e o túnel já esteja aberto por 4 horas. Quando você gira os tokens de acesso, os novos tokens gerados só podem ser usados para as 8 horas restantes.

Tópicos

- Erro de token de acesso de cliente inválido (p. 802)
- Erro de incompatibilidade de token de cliente (p. 802)
- Problemas de conectividade de dispositivo remoto (p. 803)

Erro de token de acesso de cliente inválido

Ao usar oAWS IoTtúnel seguro, você pode encontrar um erro de conexão ao usar o mesmo token de acesso do cliente (CAT) para se reconectar ao mesmo túnel. Nesse caso, o proxy local não pode se conectar ao servidor proxy de tunelamento seguro. Se você estiver usando um cliente no modo de origem, você poderá ver a seguinte mensagem de erro:

```
Invalid access token: The access token was previously used and cannot be used again
```

O erro ocorre porque o token de acesso do cliente (CAT) só pode ser usado uma vez pelo proxy local e, em seguida, ele se torna inválido. Para resolver esse erro, gire o token de acesso do cliente na`SOURCE`modo para gerar um novo CAT para a origem. Para obter um exemplo que mostra como girar o CAT de origem, consulteExemplo de GIRAR CAT de origem (p. 803).

Erro de incompatibilidade de token de cliente

> **Note**
>
> O uso de tokens de cliente para reutilizar o CAT não é recomendado. Recomendamos usar `oRotateTunnelAccessToken`API, em vez disso, para girar os tokens de acesso do cliente para se reconectar ao túnel.

Se você estiver usando tokens de cliente, poderá reutilizar o CAT para se reconectar ao túnel. Para reutilizar o CAT, você deve fornecer o token do cliente com o CAT na primeira vez em que se conectar ao tunelamento seguro. O túnel seguro armazena o token do cliente para que, para tentativas de conexão

subsequentes usando o mesmo token, o token do cliente também deve ser fornecido. Para obter mais informações sobre como usar tokens do cliente, consulte oimplementação de referência de proxy local no GitHub.

Ao usar tokens de cliente, se você estiver usando um cliente no modo de origem, poderá ver o seguinte erro:

```
Invalid client token: The provided client token does not match the client token
    that was previously set.
```

O erro ocorre porque o token do cliente fornecido não corresponde ao token do cliente fornecido com o CAT ao acessar o túnel. Para resolver esse erro, gire o CAT no`SOURCE`modo para gerar um novo CAT para a origem. Por exemplo:

Exemplo de GIRAR CAT de origem

A seguir, mostra um exemplo de como executar o`RotateTunnelAccessToken`API na`SOURCE`modo para gerar um novo CAT para a fonte:

```
aws iotsecuretunneling rotate-tunnel-access-token \
    --region <region> \
    --tunnel-id <tunnel-id> \
    --client-mode SOURCE
```

A execução desse comando gera um novo token de acesso de origem e retorna o ARN do túnel.

```
{
    "sourceAccessToken": "<source-access-token>",
    "tunnelArn": "arn:aws:iot:<region>:<account-id>:tunnel/<tunnel-id>"
}
```

Agora você pode usar o novo token de origem para conectar o proxy local no modo de origem.

```
export AWSIOT_TUNNEL_ACCESS_TOKEN=<source-access-token>
./localproxy -r <region> -s <port>
```

A seguir, um exemplo de saída da execução do proxy local:

```
...

[info]    Starting proxy in source mode
...
[info]    Successfully established websocket connection with proxy server ...
[info]    Listening for new connection on port <port>
...
```

Problemas de conectividade de dispositivo remoto

Ao usar oAWS IoTtúnel seguro, o dispositivo pode ser desconectado inesperadamente, mesmo se o túnel estiver aberto. Para identificar se um dispositivo ainda está conectado ao túnel, você pode usar oDescribeTunnelAPI ou o comandoDescribe-tunnel AWS CLI.

Um dispositivo pode ser desconectado por vários motivos. Para resolver o problema de conectividade, você pode girar o CAT no destino se o dispositivo tiver sido desconectado devido aos seguintes motivos possíveis:

- O CAT no destino tornou-se inválido.
- O token não foi entregue ao dispositivo pelo tópico MQTT reservado de tunelamento seguro:

  `$aws/things/<thing-name>/tunnels/notify`

O exemplo a seguir mostra como resolver esse problema:

Exemplo de CAT de destino de rotação

Considere um dispositivo remoto,`<RemoteThing1>`. Para abrir um túnel para essa coisa, você pode usar o seguinte comando:

```
aws iotsecuretunneling open-tunnel \
    --region <region> \
    --destination-config thingName=<RemoteThing1>,services=SSH
```

A execução desse comando gera os detalhes do túnel e o CAT para sua origem e destino.

```
{
    "sourceAccessToken": "<source-access-token>",
    "destinationAccessToken": "<destination-access-token>",
    "tunnelId": "<tunnel-id>",
    "tunnelArn": "arn:aws:iot:<region>:<account-id>:tunnel/tunnel-id"
}
```

No entanto, quando você usa oDescribeTunnelAPI, a saída indica que o dispositivo foi desconectado, conforme ilustrado abaixo:

```
aws iotsecuretunneling describe-tunnel \
    --tunnel-id <tunnel-id> \
    --region <region>
```

A execução deste comando mostra que o dispositivo ainda não está conectado.

```
{
    "tunnel": {
        ...
        "destinationConnectionState": {
            "status": "DISCONNECTED"
        },
        ...
    }
}
```

Para resolver esse erro, execute `oRotateTunnelAccessTokenAPI` com o cliente em`DESTINATIONmodo` e as configurações para o destino. A execução deste comando revoga o antigo token de acesso, gera um novo token e reenvia esse token para o tópico MQTT:

`$aws/things/<thing-name>/tunnels/notify`

```
aws iotsecuretunneling rotate-tunnel-access-token \
    --tunnel-id <tunnel-id> \
    --client-mode DESTINATION \
    --destination-config thingName=<RemoteThing1>,services=SSH \
    --region <region>
```

A execução desse comando gera o novo token de acesso, conforme mostrado abaixo. O token é então entregue ao dispositivo para se conectar ao túnel, se o agente do dispositivo estiver configurado corretamente.

```
{
    "destinationAccessToken": "destination-access-token",
    "tunnelArn": "arn:aws:iot:region:account-id:tunnel/tunnel-id"
}
```

Provisionamento de dispositivos

AWSfornece várias maneiras diferentes de provisionar um dispositivo e instalar certificados de cliente exclusivos nele. Esta seção descreve cada maneira e como selecionar o melhor para sua solução de IoT. Essas opções são descritas em detalhes no white paper intitulado,Fabricação e provisionamento de dispositivos com certificados X.509 emAWS IoT Core.

Selecione a opção que melhor se adapta à sua situação

- Você pode instalar certificados em dispositivos IoT antes que eles sejam entregues

  Se você puder instalar com segurança certificados de cliente exclusivos em seus dispositivos IoT antes que eles sejam entregues para uso pelo usuário final, você deseja usarjust-in-timeprovisionamento (JITP) (p. 815)oujust-in-timeregistro (JITR) (p. 310).

  Usando JITP e JITR, a autoridade de certificação (CA) usada para assinar o certificado do dispositivo é registrada noAWS IoTe é reconhecido porAWS IoTquando o dispositivo se conectar pela primeira vez. O dispositivo é provisionado emAWS IoTem sua primeira conexão usando os detalhes de seu modelo de provisionamento.

  Para obter mais informações sobre uma única coisa, JITP, JITR e provisionamento em massa de dispositivos que possuem certificados exclusivos, consultethe section called “Provisionamento de dispositivos com certificados de dispositivo” (p. 814).
- Usuários finais ou instaladores podem usar um aplicativo para instalar certificados em seus dispositivos IoT

  Se você não puder instalar com segurança certificados de cliente exclusivos em seu dispositivo IoT antes que eles sejam entregues ao usuário final, mas o usuário final ou um instalador puder usar um aplicativo para registrar os dispositivos e instalar os certificados de dispositivo exclusivos, você deseja usar oProvisionamento por usuário confiável (p. 810)Processo.

  Usar um usuário confiável, como um usuário final ou um instalador com uma conta conhecida, pode simplificar o processo de fabricação do dispositivo. Em vez de um certificado de cliente exclusivo, os dispositivos têm um certificado temporário que permite que o dispositivo se conecte aoAWS IoTPor apenas 5 minutos. Durante essa janela de 5 minutos, o usuário confiável obtém um certificado de cliente exclusivo com uma vida útil mais longa e o instala no dispositivo. A vida útil limitada do certificado de reivindicação minimiza o risco de um certificado comprometido.

  Para obter mais informações, consultethe section called “Provisionamento por usuário confiável” (p. 810)
- Os usuários finais NÃO PODEM usar um aplicativo para instalar certificados em seus dispositivos IoT

  Se nenhuma das opções anteriores funcionar em sua solução de IoT, oProvisionamento por reivindicação (p. 808)processo é uma opção. Com esse processo, seus dispositivos IoT têm um certificado de reivindicação compartilhado por outros dispositivos na frota. Na primeira vez que um dispositivo se conecta com um certificado de reivindicação,AWS IoTregistra o dispositivo usando seu modelo de provisionamento e emite ao dispositivo seu certificado de cliente exclusivo para acesso subsequente aoAWS IoT

  Essa opção permite o provisionamento automático de um dispositivo quando ele se conecta aAWS IoT, mas poderia apresentar um risco maior no caso de um certificado de reivindicação comprometido. Se um certificado de reivindicação ficar comprometido, você poderá desativar o certificado. A desativação do certificado de reivindicação impede que todos os dispositivos com esse certificado de reivindicação sejam registrados no future. No entanto; a desativação do certificado de reivindicação não bloqueia dispositivos que já foram provisionados.

Para obter mais informações, consultethe section called “Provisionamento por reivindicação” (p. 808)

Dispositivos de provisionamento emAWS IoT

Ao provisionar um dispositivo com o AWS IoT, você deve criar recursos para que os dispositivos e o AWS IoT possam se comunicar com segurança. Outros recursos podem ser criados para ajudar a gerenciar a frota de dispositivos. Os seguintes recursos podem ser criados durante o processo de provisionamento:

- Uma coisa do IoT.

  Coisas da IoT são entradas no registro do dispositivo do AWS IoT. Cada coisa tem um nome exclusivo e um conjunto de atributos, e está associada a um dispositivo físico. As coisas podem ser definidas usando um tipo de coisa ou agrupadas em grupos de coisas. Para obter mais informações, consulteGerenciamento de dispositivos com o AWS IoT (p. 265)

  Embora não seja necessário, criar uma coisa torna possível gerenciar a frota de dispositivos de forma mais eficaz pesquisando dispositivos por tipo de coisa, grupo de coisas e atributos de coisas. Para obter mais informações, consulteIndexação de frota (p. 836)

  > Note
  >
  > Para que o Fleet Hub indexe os dados de status de conectividade do Thing, provisione sua coisa e configure-a para que o nome da coisa corresponda ao ID do cliente usado na solicitação Connect.

- Um certificado X.509.

  Os dispositivos usam certificados X.509 para executar a autenticação mútua com o AWS IoT. É possível registrar um certificado existente ou fazer com que o AWS IoT gere e registre outro certificado para você. Associe um certificado a um dispositivo anexando-o à coisa que representa o dispositivo. Também é necessário copiar o certificado e a chave privada associada no dispositivo. Os dispositivos apresentam o certificado ao se conectar ao AWS IoT. Para obter mais informações, consulteAutenticação (p. 294)

- Uma política da IoT.

  As políticas da IoT definem as operações que um dispositivo pode executar no AWS IoT. As políticas da IoT são anexadas aos certificados do dispositivo. Quando um dispositivo apresenta o certificado ao AWS IoT, ele recebe as permissões especificadas na política. Para obter mais informações, consulteAutorização (p. 330) Cada dispositivo precisa de um certificado para se comunicar com o AWS IoT.

O AWS IoT oferece suporte ao provisionamento automatizado de frotas usando modelos de provisionamento. Os modelos de provisionamento descrevem os recursos que o AWS IoT exige para provisionar o dispositivo. Os modelos contêm variáveis que permitem usar um modelo para provisionar vários dispositivos. Ao provisionar um dispositivo, especifique valores para as variáveis específicas do dispositivo usando um dicionário ou mapa. Para provisionar outro dispositivo, especifique novos valores no dicionário.

É possível usar o provisionamento automatizado independentemente de os dispositivos terem certificados exclusivos (e a chave privada associada) ou não.

APIs de provisionamento de frota

Existem várias categorias de APIs usadas no provisionamento de frotas:

- Essas funções do plano de controle criam e gerenciam os modelos de provisionamento de frota e configuram políticas de usuário confiáveis.
  - CreateProvisioningTemplate
  - CreateProvisioningTemplateVersion
  - DeleteProvisioningTemplate
  - DeleteProvisioningTemplateVersion
  - DescribeProvisioningTemplate
  - DescribeProvisioningTemplateVersion
  - ListProvisioningTemplates
  - ListProvisioningTemplateVersions
  - UpdateProvisioningTemplate
- Usuários confiáveis podem usar essa função de plano de controle para gerar uma reivindicação de integração temporária. Essa reivindicação temporária é passada para o dispositivo durante a configuração da rede Wi-Fi ou método semelhante.
  - CreateProvisioningClaim.
- A API MQTT usada durante o processo de provisionamento por dispositivos com um certificado de reivindicação de provisionamento incorporado em um dispositivo ou passado para ele por um usuário confiável.
  - the section called “CreateCertificateFromCsr” (p. 830)
  - the section called “CreateKeysAndCertificate” (p. 832)
  - the section called “RegisterThing” (p. 833)

Provisionar dispositivos que não têm certificados de dispositivo usando o provisionamento de frotas

Usando oAWS IoTProvisionamento de frota,AWS IoTO pode gerar e entregar com segurança certificados de dispositivo e chaves privadas para seus dispositivos quando eles se conectam aoAWS IoTPela primeira vez.AWS IoTO fornece certificados de cliente que são assinados pela autoridade de certificação (CA) raiz da Amazon.

Há duas maneiras de usar o provisionamento de frotas:

- Por reivindicação
- Por usuário confiável

Provisionamento por reivindicação

Os dispositivos podem ser fabricados com um certificado de alegação de provisionamento e uma chave privada (que são credenciais de finalidade especial) incorporados neles. Se esses certificados forem registrados com o AWS IoT, o serviço poderá trocá-los por certificados de dispositivo exclusivos que o dispositivo pode usar para operações regulares. Esse processo inclui as seguintes etapas:

Antes de entregar o dispositivo

1. Chame`CreateProvisioningTemplate`Para criar um modelo de provisionamento. Essa API retorna um ARN de modelo. Para obter mais informações, consulteAPI MQTT de provisionamento de dispositivos (p. 830)

   Também é possível criar um modelo de provisionamento de frotas no console da AWS IoT.

   a. No painel de navegação, escolhaConecte-see, depois, escolhaModelos de provisionamento de frota.
   b. SelecioneCriar modeloE siga as instruções.
2. Crie certificados e chaves privadas associadas a serem usados como certificados de reivindicação de provisionamento.
3. Registre esses certificados no AWS IoT e associe uma política do IoT que restringe o uso dos certificados. O exemplo de política da IoT a seguir restringe o uso do certificado associado a essa política para dispositivos de provisionamento.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": ["iot:Connect"],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": ["iot:Publish","iot:Receive"],
            "Resource": [
                "arn:aws:iot:aws-region:aws-account-id:topic/$aws/certificates/create/
*",
                "arn:aws:iot:aws-region:aws-account-id:topic/$aws/provisioning-
templates/templateName/provision/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": "iot:Subscribe",
            "Resource": [
                "arn:aws:iot:aws-region:aws-account-id:topicfilter/$aws/certificates/
create/*",
                "arn:aws:iot:aws-region:aws-account-id:topicfilter/$aws/provisioning-
templates/templateName/provision/*"
            ]
        }
    ]
}
```

4. Conceda ao serviço do AWS IoT permissão para criar ou atualizar recursos do IoT, como coisas e certificados em sua conta ao provisionar dispositivos. Faça isso anexando a política gerenciada `AWSIoTThingsRegistration` a uma função do IAM (chamada de função de provisionamento) que confia no principal do serviço do AWS IoT.
5. Produza o dispositivo com o certificado de reivindicação de provisionamento incorporado de forma segura nele.

O dispositivo agora está pronto para ser entregue no local onde será instalado para uso.

> **Important**
>
> As chaves privadas de alegação de provisionamento devem ser protegidas o tempo todo, inclusive no dispositivo. Recomendamos usar oAWS IoT CloudWatch Métricas e logs para monitorar indicações de uso indevido. Se você detectar uso indevido, desative o certificado de alegação de provisionamento para que ele não possa ser usado para o provisionamento de dispositivos.

Como inicializar o dispositivo para uso

1. O dispositivo usa os AWS IoTSDKs de dispositivos, SDKs móveis eAWS IoTCliente de dispositivo (p. 1278) para se conectar e fazer a autenticação com o AWS IoT usando o certificado de reivindicação de provisionamento instalado no dispositivo.

   > **Note**
   >
   > Por segurança, o`certificateOwnershipToken`retornado por`CreateCertificateFromCsr (p. 830)`e`CreateKeysAndCertificate (p. 832)`expira após uma hora.`RegisterThing (p. 833)`deve ser chamado antes do`certificateOwnershipToken`Expira. Se o certificado criado por`CreateCertificateFromCsr (p. 830)`ou`CreateKeysAndCertificate (p. 832)`não foi ativado e não foi anexado a uma política ou coisa no momento em que o token expirar, o certificado é excluído. Se o token expirar, o dispositivo pode chamar`CreateCertificateFromCsr (p. 830)`ou`CreateKeysAndCertificate (p. 832)`Novamente para gerar um novo certificado.
2. O dispositivo obtém um certificado permanente e uma chave privada usando uma destas opções. O dispositivo usará o certificado e a chave para todas as futuras autenticações com o AWS IoT.

   a. Chame`CreateKeysAndCertificate` (p. 832)para criar um novo certificado e chave privada usando oAWSAutoridade certificadora.

      Ou

   b. Chame`CreateCertificateFromCsr` (p. 830)para gerar um certificado de uma solicitação de assinatura de certificado que mantém sua chave privada segura.
3. No dispositivo, chame `RegisterThing` (p. 833) para registrar o dispositivo com o AWS IoT e criar recursos de nuvem.

   O serviço de provisionamento de frotas cria recursos de nuvem, como coisas, grupos de coisas e atributos, conforme definido no modelo de provisionamento.
4. Depois de salvar o certificado permanente no dispositivo, o dispositivo deverá se desconectar da sessão iniciada com o certificado de reivindicação de provisionamento e reconectar usando o certificado permanente.

O dispositivo agora está pronto para se comunicar normalmente com o AWS IoT.

Provisionamento por usuário confiável

Em muitos casos, um dispositivo se conecta ao AWS IoT pela primeira vez quando um usuário confiável, como um usuário final ou técnico de instalação, usa um aplicativo para dispositivos móveis a fim de configurar o dispositivo em seu local implantado.

> **Important**
>
> É necessário gerenciar o acesso e a permissão do usuário confiável para realizar esse procedimento. Uma maneira de fazer isso é fornecer e manter uma conta para o usuário confiável que os autentica e concede acesso às APIs e aos recursos do AWS IoT necessários para realizar este procedimento.

Antes de entregar o dispositivo

1. Chame`CreateProvisioningTemplate`para criar um modelo de provisionamento e retornar seu*`TemplateArn`*e*`templateName`*.
2. Crie uma função do IAM que é usada por um usuário confiável para iniciar o processo de provisionamento. O modelo de provisionamento permite que somente esse usuário provisione um dispositivo. Por exemplo:

```
{
    "Effect": "Allow",
    "Action": [
        "iot:CreateProvisioningClaim"
    ],
    "Resource": [
        "arn:aws:iot:aws-region:aws-account-id:provisioningtemplate/templateName"
    ]
}
```

3. Conceda ao serviço do AWS IoT permissão para criar ou atualizar recursos do IoT, como coisas e certificados em sua conta ao provisionar dispositivos. Você pode fazer isso anexando o`AWSIoTThingsRegistration`Política gerenciada para uma função do IAM (chamada defunção de provisionamento) que confia noAWS IoTPrincipal de serviço.
4. Forneça os meios para identificar seus usuários confiáveis, como fornecendo a eles uma conta que pode autenticá-los e autorizar suas interações com oAWSAPIs necessárias para registrar seus dispositivos.

Como inicializar o dispositivo para uso

1. Um usuário confiável entra no aplicativo para dispositivos móveis ou no web service de provisionamento.
2. O aplicativo móvel ou o aplicativo web usa a função do IAM e chama`CreateProvisioningClaim`para obter um certificado de reivindicação de provisionamento temporário deAWS IoT.

   > **Note**
   >
   > Por segurança, o certificado de alegação de provisionamento temporário que`CreateProvisioningClaim`Os retornos expiram após cinco minutos. As etapas a seguir devem retornar com êxito um certificado válido antes que o certificado de reivindicação de provisionamento temporário expire. Os certificados de reivindicação de provisionamento temporário não são exibidos na lista de certificados da sua conta.

3. O aplicativo móvel ou aplicativo web fornece o certificado de reivindicação de provisionamento temporário para o dispositivo juntamente com todas as informações de configuração necessárias, como credenciais de Wi-Fi.
4. O dispositivo usa o certificado de reivindicação de provisionamento temporário para conectar-se ao AWS IoT usando o AWS IoTSDKs de dispositivos, SDKs móveis eAWS IoTCliente de dispositivo (p. 1278).
5. O dispositivo obtém um certificado permanente e uma chave privada usando uma dessas opções dentro de cinco minutos após a conexão aoAWS IoTCom o certificado de alegação de provisionamento temporário. O dispositivo usará o certificado e a chave dessas opções retornarão para todas as future autenticações comAWS IoT.

   a. Chame`CreateKeysAndCertificate` (p. 832)para criar um novo certificado e chave privada usando oAWSAutoridade certificadora.

      Ou

   b. Chame`CreateCertificateFromCsr` (p. 830)para gerar um certificado de uma solicitação de assinatura de certificado que mantém sua chave privada segura.

      > **Note**
      >
      > Lembre-se`CreateKeysAndCertificate` (p. 832)ou`CreateCertificateFromCsr` (p. 830)deve

retornar um certificado válido dentro de cinco minutos após a conexão comAWS IoTCom o certificado de alegação de provisionamento temporário.

6. O dispositivo chama `RegisterThing` (p. 833) para registrar o dispositivo com o AWS IoT e criar recursos de nuvem.

   O serviço de provisionamento de frotas cria recursos de nuvem, como coisas da IoT, grupos de coisas e atributos, conforme definido no modelo de provisionamento.

7. Depois de salvar o certificado permanente no dispositivo, o dispositivo deve se desconectar da sessão iniciada com o certificado de reivindicação de provisionamento temporário e reconectar usando o certificado permanente.

O dispositivo agora está pronto para se comunicar normalmente com o AWS IoT.

Usar ganchos de pré-provisionamento com a CLI da AWS

O procedimento a seguir cria um modelo de provisionamento com ganchos de pré-provisionamento. A função do Lambda usada aqui é um exemplo que pode ser modificado.

Como criar e aplicar um gancho de pré-provisionamento a um modelo de provisionamento

1. Crie uma função do Lambda que tenha uma entrada e saída definidas. As funções do Lambda são altamente personalizáveis`allowProvisioning`e`parameterOverrides`são necessários para criar ganchos de pré-provisionamento. Para obter mais informações sobre como criar funções do Lambda, consulteO uso doAWS Lambdacom oAWSInterface da linha de comando.

   Veja a seguir um exemplo de uma saída de função do Lambda:

```
{
  "allowProvisioning": True,
  "parameterOverrides": {
    "incomingKey0": "incomingValue0",
    "incomingKey1": "incomingValue1"
  }
}
```

2. AWS IoTO usa políticas baseadas em recursos para chamar o Lambda, portanto, é necessário concederAWS IoTpermissão para chamar sua função do Lambda.

   > **Important**
   >
   > Certifique-se de incluir o`source-arn`o`source-account`nas chaves de contexto de condição global das políticas anexadas à sua ação do Lambda para evitar a manipulação de permissão. Antes de mais informações sobre isso, consultePrevenção contra o ataque “Confused deputy” em todos os serviços (p. 342).

   Veja a seguir um exemplo de uso dopermissões de adiçãodê permissão à IoT ao seu Lambda.

```
aws lambda add-permission \
    --function-name myLambdaFunction \
    --statement-id iot-permission \
    --action lambda:InvokeFunction \
    --principal iot.amazonaws.com
```

3. Adicione um gancho de pré-provisionamento a um modelo usando o comando create-provisioning-template ou update-provisioning-template.

O exemplo de CLI a seguir usa o create-provisioning-template para criar um modelo de provisionamento que tenha ganchos de pré-provisionamento:

```
aws iot create-provisioning-template \
    --template-name myTemplate \
    --provisioning-role-arn arn:aws:iam:us-east-1:1234564789012:role/myRole \
    --template-body file://template.json \
    --pre-provisioning-hook file://hooks.json
```

A saída desse comando é semelhante à seguinte:

```
{
    "templateArn": "arn:aws:iot:us-east-1:1234564789012:provisioningtemplate/
myTemplate",
    "defaultVersionId": 1,
    "templateName": myTemplate
}
```

Também é possível carregar um parâmetro de um arquivo em vez de digitar tudo como um valor de parâmetro da linha de comando para economizar tempo. Para obter mais informações, consulteCarregarAWS CLIParâmetros de um arquivo. Veja a seguir o parâmetro `template` no formato JSON expandido:

```
{
    "Parameters" : {
        "DeviceLocation": {
            "Type": "String"
        }
    },
    "Mappings": {
        "LocationTable": {
            "Seattle": {
                "LocationUrl": "https://example.aws"
            }
        }
    },
    "Resources" : {
        "thing" : {
            "Type" : "AWS::IoT::Thing",
            "Properties" : {
                "AttributePayload" : {
                    "version" : "v1",
                    "serialNumber" : "serialNumber"
                },
                "ThingName" : {"Fn::Join":["",["ThingPrefix_",
{"Ref":"SerialNumber"}]]},
                "ThingTypeName" : {"Fn::Join":["",["ThingTypePrefix_",
{"Ref":"SerialNumber"}]]},
                "ThingGroups" : ["widgets", "WA"],
                "BillingGroup": "BillingGroup"
            },
            "OverrideSettings" : {
                "AttributePayload" : "MERGE",
                "ThingTypeName" : "REPLACE",
                "ThingGroups" : "DO_NOTHING"
            }
        },
        "certificate" : {
            "Type" : "AWS::IoT::Certificate",
            "Properties" : {
                "CertificateId": {"Ref": "AWS::IoT::Certificate::Id"},
```

```
                "Status" : "Active"
            }
        },
        "policy" : {
            "Type" : "AWS::IoT::Policy",
            "Properties" : {
                "PolicyDocument" : {
                    "Version": "2012-10-17",
                    "Statement": [{
                        "Effect": "Allow",
                        "Action":["iot:Publish"],
                        "Resource": ["arn:aws:iot:us-east-1:504350838278:topic/foo/
bar"]
                    }]
                }
            }
        }
    },
    "DeviceConfiguration": {
        "FallbackUrl": "https://www.example.com/test-site",
        "LocationUrl": {
            "Fn::FindInMap": ["LocationTable",{"Ref": "DeviceLocation"},
 "LocationUrl"]}
    }
}
```

Veja a seguir o parâmetro `pre-provisioning-hook` no formato JSON expandido:

```
{
     "targetArn" : "arn:aws:lambda:us-
east-1:765219403047:function:pre_provisioning_test",
     "payloadVersion" : "2020-04-01"
}
```

Provisionamento de dispositivos com certificados de dispositivo

O AWS IoT fornece três maneiras de provisionar dispositivos quando eles já têm um certificado de dispositivo (e uma chave privada associada) neles:

- Provisionamento de uma única coisa com um modelo de provisionamento. Essa é uma boa opção se você precisa provisionar somente um dispositivo por vez.
- O provisionamento just-in-time (JITP) com um modelo que provisiona um dispositivo ao se conectar ao AWS IoT pela primeira vez. Essa é uma boa opção se você precisa registrar uma grande quantidade de dispositivos, mas não tem informações sobre eles para montar uma lista de provisionamento em massa.
- Registro em massa. Essa opção permite que você especifique uma lista de valores de um modelo de provisionamento de uma única coisa que são armazenados em um arquivo em um bucket do S3. Essa abordagem funciona bem se você tem uma grande quantidade de dispositivos conhecidos cujas características desejadas podem ser montadas por você em uma lista.

Tópicos

- Provisionamento de uma única coisa (p. 815)
- Provisionamento just-in-time (p. 815)
- Registro em massa (p. 818)

Provisionamento de uma única coisa

Para provisionar uma coisa, use a API RegisterThing ou o comando `register-thing` da CLI. O comando `register-thing` da CLI usa os seguintes argumentos:

--template-body

O modelo provisionado.

--parameters

Uma lista de pares de nome-valor para os parâmetros usados no modelo de provisionamento, no formato JSON (por exemplo, `{"ThingName" : "MyProvisionedThing", "CSR" : "csr-text"}`).

Consulte Modelos de provisionamento (p. 819).

RegisterThing ou `register-thing` retorna os ARNs dos recursos e o texto do certificado que criou:

```
{
    "certificatePem": "certificate-text",
    "resourceArns": {
    "PolicyLogicalName": "arn:aws:iot:us-
west-2:123456789012:policy/2A6577675B7CD1823E271C7AAD8184F44630FFD7",
    "certificate": "arn:aws:iot:us-west-2:123456789012:cert/
cd82bb924d4c6ccbb14986dcb4f40f30d892cc6b3ce7ad5008ed6542eea2b049",
    "thing": "arn:aws:iot:us-west-2:123456789012:thing/MyProvisionedThing"
    }
}
```

Se um parâmetro for omitido do dicionário, o valor padrão será usado. Se nenhum valor padrão estiver especificado, o parâmetro não será substituído por um valor.

Provisionamento just-in-time

Você pode ter seus dispositivos provisionados quando eles tentarem se conectar à pela primeira vezAWS IoTcom just-in-time provisionamento (JITP). Para provisionar o dispositivo, você deve habilitar o registro automático e associar um modelo de provisionamento ao certificado da CA usado para assinar o certificado do dispositivo. Os sucessos e erros de provisionamento são registrados comoMétricas de provisionamento de dispositivos (p. 451)No Amazon CloudWatch.

Você pode definir essas configurações ao registrar um novo certificado da CA com a API RegisterCACertificate ou o comando `register-ca-certificate` da CLI:

```
aws iot register-ca-certificate --ca-certificate file://your-ca-cert --verification-cert
                file://your-verification-cert --set-as-active --allow-auto-registration --
registration-config file://your-template
```

Para obter mais informações, consulte Registrar um certificado CA.

Também é possível usar a API UpdateCACertificate ou o comando `update-ca-certificate` da CLI para atualizar as configurações de um certificado CA:

```
aws iot update-ca-certificate --certificate-id caCertificateId --new-auto-registration-
status ENABLE --registration-config file://your-template
```

> **Note**
>
> Provisionamento just-in-time JITP chama outrosAWS IoTAs APIs do plano de controle durante o processo de provisionamento. Essas chamadas podem exceder oAWS IoTCotas de controle de utilização doDefina para sua conta e resulte em chamadas limitadas. ContatoAWSSupport ao clientepara aumentar as cotas do controle de utilização, se necessário.

Quando um dispositivo tenta se conectar ao AWS IoT usando um certificado assinado por um certificado CA registrado, o AWS IoT carrega o modelo a partir do certificado CA e usa-o para chamar RegisterThing (p. 833). O fluxo de trabalho de JITP registra primeiro um certificado com um valor de status de PENDING_ACTIVATION. Quando o fluxo de provisionamento do dispositivo for concluído, o status do certificado será alterado para ACTIVE.

A AWS IoT define os seguintes parâmetros que você pode declarar e fazer referência nos modelos de provisionamento:

- `AWS::IoT::Certificate::Country`
- `AWS::IoT::Certificate::Organization`
- `AWS::IoT::Certificate::OrganizationalUnit`
- `AWS::IoT::Certificate::DistinguishedNameQualifier`
- `AWS::IoT::Certificate::StateName`
- `AWS::IoT::Certificate::CommonName`
- `AWS::IoT::Certificate::SerialNumber`
- `AWS::IoT::Certificate::Id`

Os valores para esses parâmetros de modelo de provisionamento são limitados ao que o JITP pode extrair do campo de assunto do certificado do dispositivo que está sendo provisionado. O certificado deve conter valores para todos os parâmetros no corpo do modelo. O parâmetro `AWS::IoT::Certificate::Id` se refere a um ID gerado internamente, e não a um ID contido no certificado. Você pode obter o valor desse ID usando a função `principal()` em uma regra da AWS IoT.

> **Note**
>
> Você pode provisionar dispositivos usando oAWS IoT Core just-in-time recurso de provisionamento (JITP) sem precisar enviar toda a cadeia de confiança na primeira conexão dos dispositivos comAWS IoT Core. A apresentação do certificado da CA é opcional, mas o dispositivo é necessário para enviar oSNI (Server Name Indication)extensão quando eles se conectam.

O arquivo JSON a seguir é um exemplo de modelo de JITP completo. O valor do campo `templateBody` deve ser um objeto JSON especificado como uma string de escape e pode usar somente os valores da lista anterior. É possível usar uma variedade de ferramentas para criar a saída JSON necessária, como `json.dumps` (Python) ou `JSON.stringify` (Node). O valor do campo `roleARN` deve ser o ARN de uma função que tenha a `AWSIoTThingsRegistration` anexada a ele. Além disso, o modelo pode usar um `PolicyName` existente em vez da `PolicyDocument` em linha no exemplo. (O primeiro exemplo adiciona quebras de linha para legibilidade, mas você pode copiar e colar o modelo que é exibido diretamente abaixo dele.)

```
        {
            "templateBody" : "{
                \r\n    \"Parameters\" : {\r\n
                            \"AWS::IoT::Certificate::CommonName\": {\r\n    \"Type\":
 \"String\"\r\n        },\r\n
                            \"AWS::IoT::Certificate::SerialNumber\": {\r\n  \"Type\":
 \"String\"\r\n        },\r\n
                            \"AWS::IoT::Certificate::Country\": {\r\n   \"Type\": \"String
\"\r\n        },\r\n
```

```
                            \"AWS::IoT::Certificate::Id\": {\r\n    \"Type\": \"String\"\r
\n        }\r\n    },\r\n
                        \"Resources\": {\r\n
                            \"thing\": {\r\n
                                \"Type\": \"AWS::IoT::Thing\",\r\n
                                \"Properties\": {\r\n
                                    \"ThingName\": {\r\n    \"Ref\":
 \"AWS::IoT::Certificate::CommonName\"\r\n  },\r\n
                                    \"AttributePayload\": {\r\n
                                        \"version\": \"v1\",\r\n
                                        \"serialNumber\": {\r\n
                                            \"Ref\": \"AWS::IoT::Certificate::SerialNumber
\"\r\n    }\r\n   },\r\n
                                    \"ThingTypeName\": \"lightBulb-versionA\",\r\n

                                    \"ThingGroups\": [\r\n
                                        \"v1-lightbulbs\",\r\n  {\r\n

                                            \"Ref\": \"AWS::IoT::Certificate::Country\"\r
\n }\r\n   ]\r\n   },\r\n
                            \"OverrideSettings\": {\r\n
                                \"AttributePayload\": \"MERGE\",\r\n
                                \"ThingTypeName\": \"REPLACE\",\r\n
                                \"ThingGroups\": \"DO_NOTHING\"\r\n }\r\n   },\r\n
                        \"certificate\": {\r\n
                            \"Type\": \"AWS::IoT::Certificate\",\r\n
                            \"Properties\": {\r\n
                                \"CertificateId\": {\r\n    \"Ref\":
 \"AWS::IoT::Certificate::Id\"\r\n  },\r\n
                                \"Status\": \"ACTIVE\"\r\n  },\r\n
                            \"OverrideSettings\": {\r\n
                                \"Status\": \"DO_NOTHING\"\r\n  }\r\n   },\r\n
                        \"policy\": {\r\n
                            \"Type\": \"AWS::IoT::Policy\",\r\n
                            \"Properties\": {\r\n
                                \"PolicyDocument\": \"{
                                    \\\"Version\\\": \\\"2012-10-17\\\",
                                    \\\"Statement\\\": [{
                                    \\\"Effect\\\": \\\"Allow\\\",
                                    \\\"Action\\\":[\\\"iot:Publish\\\"],
                                    \\\"Resource\\\": [\\\"arn:aws:iot:us-
east-1:123456789012:topic\/sample\/topic\\\"] }] }\"\r\n   }\r\n   }\r\n
                            }\r\n}",
           "roleArn" : "arn:aws:iam::123456789012:role/Provisioning-JITP"
        }
```

Veja a versão que você pode copiar e colar:

```
    {
      "templateBody" : "{\r\n    \"Parameters\" : {\r\n
 \"AWS::IoT::Certificate::CommonName\": {\r\n            \"Type\": \"String\"\r\n        },
\r\n        \"AWS::IoT::Certificate::SerialNumber\": {\r\n            \"Type\": \"String
\"\r\n        },\r\n        \"AWS::IoT::Certificate::Country\": {\r\n            \"Type\":
 \"String\"\r\n        },\r\n        \"AWS::IoT::Certificate::Id\": {\r\n            \"Type
\": \"String\"\r\n        }\r\n    },\r\n    \"Resources\": {\r\n        \"thing\": {\r\n
            \"Type\": \"AWS::IoT::Thing\",\r\n            \"Properties\": {\r\n
      \"ThingName\": {\r\n                    \"Ref\": \"AWS::IoT::Certificate::CommonName
\"\r\n                },\r\n                \"AttributePayload\": {\r\n
   \"version\": \"v1\",\r\n                    \"serialNumber\": {\r\n
      \"Ref\": \"AWS::IoT::Certificate::SerialNumber\"\r\n                    }\r\n
           },\r\n                \"ThingTypeName\": \"lightBulb-versionA\",\r\n
      \"ThingGroups\": [\r\n                        \"v1-lightbulbs\",\r\n
 {\r\n                        \"Ref\": \"AWS::IoT::Certificate::Country\"\r\n
```

```
        }\r\n                ]\r\n            },\r\n            \"OverrideSettings\": {\r
\n                \"AttributePayload\": \"MERGE\",\r\n                \"ThingTypeName\":
 \"REPLACE\",\r\n                \"ThingGroups\": \"DO_NOTHING\"\r\n            }\r\n
   },\r\n        \"certificate\": {\r\n            \"Type\": \"AWS::IoT::Certificate\",\r\n
            \"Properties\": {\r\n                \"CertificateId\": {\r\n
   \"Ref\": \"AWS::IoT::Certificate::Id\"\r\n                },\r\n                \"Status
\": \"ACTIVE\"\r\n            },\r\n            \"OverrideSettings\": {\r\n
  \"Status\": \"DO_NOTHING\"\r\n            }\r\n        },\r\n        \"policy\": {\r\n
           \"Type\": \"AWS::IoT::Policy\",\r\n            \"Properties\": {\r\n
      \"PolicyDocument\": \"{ \\\"Version\\\": \\\"2012-10-17\\\", \\\"Statement\\\": [{ \
\\\"Effect\\\": \\\"Allow\\\", \\\"Action\\\":[\\\"iot:Publish\\\"], \\\"Resource\\\": [\\
\"arn:aws:iot:us-east-1:123456789012:topic\/foo\/bar\\\"] }] }\"\r\n            }\r\n
  }\r\n    }\r\n}",
      "roleArn" : "arn:aws:iam::123456789012:role/JITPRole"
    }
```

Este modelo de exemplo declara valores para os parâmetros de provisionamento `AWS::IoT::Certificate::CommonName`, `AWS::IoT::Certificate::SerialNumber`, `AWS::IoT::Certificate::Country` e `AWS::IoT::Certificate::Id` que são extraídos do certificado e usados na seção `Resources`. Em seguida, o fluxo de trabalho de JITP usa esse modelo para executar as seguintes ações:

- Registrar um certificado e definir seu status como PENDING_ACTIVE.
- Criar um recurso de coisa.
- Criar um recurso de política.
- Anexar a política ao certificado.
- Anexar um certificado à coisa.
- Atualizar o status do certificado como ACTIVE.

Observe que o provisionamento do dispositivo falhará se o certificado não tiver todas as propriedades mencionadas na`Parameters`Seção do do`templateBody`. Por exemplo, se`AWS::IoT::Certificate::Country`está incluído no modelo, mas o certificado não tem um`Country`propriedade, o provisionamento do dispositivo falha.

Você também pode usar CloudTrail Para solucionar problemas com seu modelo de JITP. Para obter informações sobre as métricas registradas no Amazon CloudWatch, consulteMétricas de provisionamento de dispositivos (p. 451).

Registro em massa

Você pode usar o`start-thing-registration-task`Comando registrar coisas em massa. Esse comando usa um modelo de registro, um nome de bucket do S3, um nome de chave, e o ARN de uma função que concede acesso ao arquivo no bucket do S3. O arquivo no bucket do S3 contém os valores usados para substituir os parâmetros no modelo. O arquivo deve ser um JSON delimitado por nova linha. Cada linha contém todos os valores dos parâmetros para o registro de um único dispositivo. Por exemplo:

```
{"ThingName": "foo", "SerialNumber": "123", "CSR": "csr1"}
{"ThingName": "bar", "SerialNumber": "456", "CSR": "csr2"}
```

As seguintes APIs relacionadas ao registro em massa podem ser úteis:

- ListThingRegistrationTasks: Lista as tarefas atuais de provisionamento de coisas em massa.
- DescribeThingRegistrationTask: Fornece informações sobre uma tarefa específica de registro de coisas em massa.
- StopThingRegistrationTask: Interrompe uma tarefa de registro de coisas em massa.

- ListThingRegistrationTaskReports: Usada para verificar os resultados e as falhas de uma tarefa de registro de coisas em massa.

  Note

  - Somente uma tarefa de operação de registro de coisas em massa pode ser executada de cada vez (por conta).
  - Operações de registro em massa chamam outras APIs do plano de controle da AWS IoT. Essas chamadas podem exceder oAWS IoTCotas de controle de utilização doEm sua conta e causa erros de limitação. ContatoAWSSupport ao clientepara aumentar o seuAWS IoTCotas de limitação, se necessário.

Modelos de provisionamento

Um modelo de provisionamento é um documento JSON que usa parâmetros para descrever os recursos que seu dispositivo deve usar para interagir com a AWS IoT. Um modelo contém duas seções: `Parameters` e `Resources`. Existem dois tipos de modelos de provisionamento na AWS IoT. Um é usado para just-in-time provisionamento (JITP) e registro em massa e o segundo é usado para provisionamento de frotas.

Seção de parâmetros

A seção `Parameters` declara os parâmetros usados na seção `Resources`. Cada parâmetro declara um nome, um tipo e um valor padrão opcional. O valor padrão é usado quando o dicionário passado com o modelo não contém um valor para o parâmetro. A seção `Parameters` de um documento de modelo é semelhante à seguinte:

```
{
    "Parameters" : {
        "ThingName" : {
            "Type" : "String"
        },
        "SerialNumber" : {
            "Type" : "String"
        },
        "Location" : {
            "Type" : "String",
            "Default" : "WA"
        },
        "CSR" : {
            "Type" : "String"
        }
    }
}
```

Esse trecho de código do modelo declara quatro parâmetros: `ThingName`, `SerialNumber`, `Location` e `CSR`. Todos esses parâmetros são do tipo `String`. O parâmetro `Location` declara um valor padrão de `"WA"`.

Seção de recursos

A seção `Resources` do modelo declara os recursos necessários para o dispositivo se comunicar com a AWS IoT: uma coisa, um certificado e uma ou mais políticas da IoT. Cada recurso especifica um nome lógico, um tipo e um conjunto de propriedades.

Um nome lógico permite que você faça referência a um recurso em outro lugar no modelo.

O tipo especifica o tipo de recurso que você está declarando. Os tipos válidos são:

- `AWS::IoT::Thing`
- `AWS::IoT::Certificate`
- `AWS::IoT::Policy`

As propriedades que você especifica dependem do tipo de recurso que você está declarando.

Recursos de coisas

Os recursos de coisas são declarados usando as seguintes propriedades:

- `ThingName`: String.
- `AttributePayload`: Optional. Uma lista de pares nome-valor.
- `ThingTypeName`: Optional. Sequência para um tipo de coisa associado à coisa.
- `ThingGroups`: Optional. Uma lista de grupos aos quais a coisa pertence.

Recursos de certificados

É possível especificar certificados de uma das seguintes maneiras:

- Uma solicitação de assinatura de certificado (CSR).
- Um ID de certificado de um certificado de dispositivo existente. (Somente IDs de certificado podem ser usados com um modelo de provisionamento de frotas.)
- Um certificado de dispositivo criado com um certificado da CA registrado na AWS IoT. Se houver mais de um certificado CA registrado com o mesmo campo de assunto, você também deverá passar o certificado CA usado para assinar o certificado do dispositivo.

> **Note**
>
> Ao declarar um certificado em um modelo, use somente um desses métodos. Por exemplo, se você usar uma CSR, não será possível especificar também um ID de certificado ou um certificado de dispositivo. Para obter mais informações, consulteCertificados do cliente X.509 (p. 298)

Para obter mais informações, consulteVisão geral do certificado X.509 (p. 294)

Os recursos de certificados são declarados usando as seguintes propriedades:

- `CertificateSigningRequest`: String.
- `CertificateId`: String.
- `CertificatePem`: String.
- `CACertificatePem`: String.
- `Status`: Optional. String que pode ser `ACTIVE` ou `INACTIVE`. Padronizada como ACTIVE.

Exemplos:

- Certificado especificado com um CSR:

```
{
    "certificate" : {
        "Type" : "AWS::IoT::Certificate",
        "Properties" : {
            "CertificateSigningRequest": {"Ref" : "CSR"},
```

```
            "Status" : "ACTIVE"
        }
    }
}
```

- Certificado especificado com um ID de certificado existente:

```
{
    "certificate" : {
        "Type" : "AWS::IoT::Certificate",
        "Properties" : {
            "CertificateId": {"Ref" : "CertificateId"}
        }
    }
}
```

- Certificado especificado com um certificado .pem existente .pem e certificado .pem da CA:

```
{
    "certificate" : {
        "Type" : "AWS::IoT::Certificate",
        "Properties" : {
            "CACertificatePem": {"Ref" : "CACertificatePem"},
            "CertificatePem": {"Ref" : "CertificatePem"}
        }
    }
}
```

Recursos de políticas

Os recursos de políticas são declarados com uma das seguintes propriedades:

- `PolicyName`: Optional. String. Padroniza para um hash do documento de política. O`PolicyName`só pode fazer referênciaAWS IoTpolíticas, mas não políticas do IAM. Se você estiver usando uma política do AWS IoT existente, insira o nome da política para a propriedade `PolicyName`. Não inclua a propriedade `PolicyDocument`.
- `PolicyDocument`: Optional. Um objeto JSON especificado como uma string de escape. Se `PolicyDocument` não for fornecido, a política já deverá estar criada.

> **Note**
>
> Se uma seção `Policy` estiver presente, `PolicyName` ou `PolicyDocument`, mas não ambos, deve ser especificado.

Configurações de substituição

Se um modelo especificar um recurso que já existe, a seção `OverrideSettings` permitirá que você especifique a ação a ser executada:

`DO_NOTHING`

Deixe o recurso como está.

`REPLACE`

Substitui o recurso pelo recurso especificado no modelo.

`FAIL`

Faz com que a solicitação falhe com uma `ResourceConflictsException`.

`MERGE`

Válido apenas para as propriedades `ThingGroups` e `AttributePayload` de uma `thing`. Mescla os atributos ou associações de grupo existentes da coisa com os especificados no modelo.

Ao declarar um recurso de coisa, você pode especificar `OverrideSettings` para as seguintes propriedades:

- `ATTRIBUTE_PAYLOAD`
- `THING_TYPE_NAME`
- `THING_GROUPS`

Ao declarar um recurso de certificado, você pode especificar `OverrideSettings` para a propriedade `Status`.

`OverrideSettings` não estão disponíveis para recursos de política.

Exemplo de recurso

O trecho de código do modelo a seguir declara uma coisa, um certificado e uma política:

```
{
    "Resources" : {
        "thing" : {
            "Type" : "AWS::IoT::Thing",
            "Properties" : {
                "ThingName" : {"Ref" : "ThingName"},
                "AttributePayload" : { "version" : "v1", "serialNumber" :  {"Ref" :
 "SerialNumber"}},
                "ThingTypeName" :  "lightBulb-versionA",
                "ThingGroups" : ["v1-lightbulbs", {"Ref" : "Location"}]
            },
            "OverrideSettings" : {
                "AttributePayload" : "MERGE",
                "ThingTypeName" : "REPLACE",
                "ThingGroups" : "DO_NOTHING"
            }
        },
        "certificate" : {
            "Type" : "AWS::IoT::Certificate",
            "Properties" : {
                "CertificateSigningRequest": {"Ref" : "CSR"},
                "Status" : "ACTIVE"
            }
        },
        "policy" : {
            "Type" : "AWS::IoT::Policy",
            "Properties" : {
                "PolicyDocument" : "{ \"Version\": \"2012-10-17\", \"Statement\":
 [{ \"Effect\": \"Allow\", \"Action\":[\"iot:Publish\"], \"Resource\": [\"arn:aws:iot:us-
east-1:123456789012:topic/foo/bar\"] }] }"
            }
        }
    }
}
```

A coisa é declarada com:

- O nome lógico `"thing"`.

- O tipo `AWS::IoT::Thing`.
- Um conjunto de propriedades de coisas.

  As propriedades da coisa incluem o nome, um conjunto de atributos, um nome opcional de tipo de coisa e uma lista opcional de grupos de coisas aos quais a coisa pertence.

Os parâmetros são referenciados por `{"Ref":"parameter-name"}`. Quando o modelo é avaliado, os parâmetros são substituídos pelo valor do parâmetro do dicionário passado com o modelo.

O certificado é declarado com:

- O nome lógico `"certificate"`.
- O tipo `AWS::IoT::Certificate`.
- Um conjunto de propriedades.

  As propriedades incluem a CSR do certificado e a configuração do status como `ACTIVE`. O texto da CSR é passado como um parâmetro no dicionário passado com o modelo.

A política é declarada com:

- O nome lógico `"policy"`.
- O tipo `AWS::IoT::Policy`.
- O nome de uma política existente ou um documento de política.

Exemplo de modelo para JITP e registro em massa

O arquivo JSON a seguir é um exemplo de modelo de provisionamento completo que especifica o certificado com uma CSR:

(O valor do campo `PolicyDocument` deve ser um objeto JSON especificado como uma string de escape.)

```
{
    "Parameters" : {
        "ThingName" : {
            "Type" : "String"
        },
        "SerialNumber" : {
            "Type" : "String"
        },
        "Location" : {
            "Type" : "String",
            "Default" : "WA"
        },
        "CSR" : {
            "Type" : "String"
        }
    },
    "Resources" : {
        "thing" : {
            "Type" : "AWS::IoT::Thing",
            "Properties" : {
                "ThingName" : {"Ref" : "ThingName"},
                "AttributePayload" : { "version" : "v1", "serialNumber" :  {"Ref" :
 "SerialNumber"}},
                "ThingTypeName" :  "lightBulb-versionA",
                "ThingGroups" : ["v1-lightbulbs", {"Ref" : "Location"}]
            }
        },
```

```
        "certificate" : {
            "Type" : "AWS::IoT::Certificate",
            "Properties" : {
                "CertificateSigningRequest": {"Ref" : "CSR"},
                "Status" : "ACTIVE"
            }
        },
        "policy" : {
            "Type" : "AWS::IoT::Policy",
            "Properties" : {
                "PolicyDocument" : "{ \"Version\": \"2012-10-17\", \"Statement\":
 [{ \"Effect\": \"Allow\", \"Action\":[\"iot:Publish\"], \"Resource\": [\"arn:aws:iot:us-
east-1:123456789012:topic/foo/bar\"] }] }"
            }
        }
    }
}
```

O arquivo JSON a seguir é um exemplo de modelo de provisionamento completo que especifica um certificado existente com um ID de certificado:

```
{
    "Parameters" : {
        "ThingName" : {
            "Type" : "String"
        },
        "SerialNumber" : {
            "Type" : "String"
        },
        "Location" : {
            "Type" : "String",
            "Default" : "WA"
        },
        "CertificateId" : {
            "Type" : "String"
        }
    },
    "Resources" : {
        "thing" : {
            "Type" : "AWS::IoT::Thing",
            "Properties" : {
                "ThingName" : {"Ref" : "ThingName"},
                "AttributePayload" : { "version" : "v1", "serialNumber" :  {"Ref" :
 "SerialNumber"}},
                "ThingTypeName" :  "lightBulb-versionA",
                "ThingGroups" : ["v1-lightbulbs", {"Ref" : "Location"}]
            }
        },
        "certificate" : {
            "Type" : "AWS::IoT::Certificate",
            "Properties" : {
                "CertificateId": {"Ref" : "CertificateId"}
            }
        },
        "policy" : {
            "Type" : "AWS::IoT::Policy",
            "Properties" : {
                "PolicyDocument" : "{ \"Version\": \"2012-10-17\", \"Statement\":
 [{ \"Effect\": \"Allow\", \"Action\":[\"iot:Publish\"], \"Resource\": [\"arn:aws:iot:us-
east-1:123456789012:topic/foo/bar\"] }] }"
            }
        }
    }
}
```

Provisionamento de frota

Os modelos de provisionamento de frotas são usados pela AWS IoT para configurar a nuvem e a configuração do dispositivo. Esses modelos usam os mesmos parâmetros e recursos que o JITP e os modelos de registro em massa. Para obter mais informações, consulteModelos de provisionamento (p. 819) Os modelos de provisionamento de frotas podem conter uma seção `Mapping` e uma seção `DeviceConfiguration`. É possível usar funções intrínsecas dentro de um modelo de provisionamento de frotas para gerar uma configuração específica do dispositivo Os modelos de provisionamento de frotas são recursos nomeados e são identificados por ARNs (por exemplo, `arn:aws:iot:us-west-2:1234568788:provisioningtemplate/`*`templateName`*).

Mapeamentos

A seção opcional `Mappings` corresponde uma chave a um conjunto de valores nomeados correspondente. Por exemplo, se você quiser definir valores com base em umAWSRegião, você pode criar um mapeamento que usa oRegião da AWSnome como uma chave e contém os valores que você deseja especificar para cada região específica. Você usa a função intrínseca `Fn::FindInMap` para recuperar valores em um mapa.

Não é possível incluir parâmetros, pseudoparâmetros ou chamar funções intrínsecas na seção `Mappings`.

Configuração do dispositivo

A seção de configuração do dispositivo contém dados arbitrários que você deseja enviar para os dispositivos ao provisionar. Por exemplo:

```
{
    "DeviceConfiguration": {
        "Foo":"Bar"
    }
}
```

Se você estiver enviando mensagens para seus dispositivos usando o JavaScript Formato da carga da Object Notation (JSON)AWS IoT Coreformata esses dados como JSON. Se você estiver usando o formato CBOR (Concise Binary Object Representation),AWS IoT Coreformata esses dados como CBOR. O`DeviceConfiguration`A seção não oferece suporte a objetos JSON aninhados.

Funções intrínsecas

As funções intrínsecas são usadas em qualquer seção do modelo de provisionamento, exceto a seção `Mappings`.

`Fn::Join`

Anexa um conjunto de valores em um único valor, separados pelo delimitador especificado. Se um delimitador é a string vazia, o conjunto de valores é concatenado sem delimitador.

`Fn::Select`

Retorna um único objeto de uma lista de objetos por índice.

> **Important**
>
> `Fn::Select` não verifica valores `null` ou se o índice está fora dos limites da matriz. Ambas as condições resultam em um erro de provisionamento, portanto, você deve garantir que escolheu um valor de índice válido e que a lista contém valores não nulos.

`Fn::FindInMap`

Retorna o valor correspondente às chaves em um mapa de dois níveis que é declarado na seção `Mappings`.

`Fn::Split`

Divide uma string em uma lista de valores de string para que seja possível selecionar um elemento na lista de strings. Especifique um delimitador que determina onde a string é dividida (por exemplo, uma vírgula). Depois de dividir uma string, use `Fn::Select` para selecionar um elemento.

Por exemplo, se uma string de IDs de sub-rede delimitada por vírgulas for importada para seu modelo de pilha, você poderá dividir a string em cada vírgula. Na lista de IDs de sub-rede, use `Fn::Select` para especificar um ID de sub-rede para um recurso.

`Fn::Sub`

Substitui variáveis em uma string de entrada por valores especificados por você. É possível usar essa função para criar comandos ou saídas que incluem valores que não estão disponíveis até que você crie ou atualize uma pilha.

Exemplo de modelo de provisionamento de frota

```
{
    "Parameters" : {
        "ThingName" : {
            "Type" : "String"
        },
        "SerialNumber": {
            "Type": "String"
        },
        "DeviceLocation": {
            "Type": "String"
        }
    },
    "Mappings": {
        "LocationTable": {
            "Seattle": {
                "LocationUrl": "https://example.aws"
            }
        }
    },
    "Resources" : {
        "thing" : {
            "Type" : "AWS::IoT::Thing",
            "Properties" : {
                "AttributePayload" : {
                    "version" : "v1",
                    "serialNumber" : "serialNumber"
                },
                "ThingName" : {"Ref" : "ThingName"},
                "ThingTypeName" : {"Fn::Join":["",["ThingPrefix_",
{"Ref":"SerialNumber"}]]},
                "ThingGroups" : ["v1-lightbulbs", "WA"],
                "BillingGroup": "LightBulbBillingGroup"
            },
            "OverrideSettings" : {
                "AttributePayload" : "MERGE",
                "ThingTypeName" : "REPLACE",
                "ThingGroups" : "DO_NOTHING"
            }
        },
        "certificate" : {
```

```
            "Type" : "AWS::IoT::Certificate",
            "Properties" : {
                "CertificateId": {"Ref": "AWS::IoT::Certificate::Id"},
                "Status" : "Active"
            }
        },
        "policy" : {
            "Type" : "AWS::IoT::Policy",
            "Properties" : {
                "PolicyDocument" : {
                    "Version": "2012-10-17",
                    "Statement": [{
                        "Effect": "Allow",
                        "Action":["iot:Publish"],
                        "Resource": ["arn:aws:iot:us-east-1:123456789012:topic/foo/bar"]
                    }]
                }
            }
        }
    },
    "DeviceConfiguration": {
        "FallbackUrl": "https://www.example.com/test-site",
        "LocationUrl": {
            "Fn::FindInMap": ["LocationTable",{"Ref": "DeviceLocation"}, "LocationUrl"]}
        }
}
```

> **Note**
>
> Um modelo de provisionamento existente pode ser atualizado para adicionar um gancho de pré-provisionamento (p. 827).

Ganchos de pré-provisionamento

AWSA recomenda o uso de funções de gancho de pré-provisionamento ao criar modelos de provisionamento para permitir mais controle sobre quais e quantos dispositivos são incorporados à conta. Os ganchos de pré-provisionamento são funções do Lambda que validam parâmetros passados do dispositivo antes de permitir que o dispositivo seja provisionado. Essa função do Lambda deve existir na sua conta antes de provisionar um dispositivo, pois ela é chamada sempre que um dispositivo envia uma solicitação por meio dethe section called “RegisterThing” (p. 833).

> **Important**
>
> Certifique-se de incluir `osource-arnousource-accountnas` chaves de contexto de condição global das políticas anexadas à sua ação do Lambda para evitar a manipulação de permissão. Antes de mais informações sobre isso, consultePrevenção contra o ataque “Confused deputy” em todos os serviços (p. 342).

Para dispositivos a serem provisionados, a função do Lambda deve aceitar o objeto de entrada e retornar o objeto de saída descrito nesta seção. O provisionamento prosseguirá somente se a função do Lambda retornar um objeto com`"allowProvisioning": True`.

Entrada de gancho de pré-provisão

AWS IoTO envia esse objeto para a função do Lambda quando um dispositivo é registrado noAWS IoT.

```
{
    "claimCertificateId" : "string",
    "certificateId" : "string",
```

```
    "certificatePem" : "string",
    "templateArn" : "arn:aws:iot:us-east-1:1234567890:provisioningtemplate/MyTemplate",
    "clientId" : "221a6d10-9c7f-42f1-9153-e52e6fc869c1",
    "parameters" : {
        "string" : "string",
        ...
    }
}
```

O`parameters`O objeto passado para a função do Lambda contém as propriedades na`parameters`argumento passado nothe section called “RegisterThing” (p. 833)Carga da solicitação.

Valor de retorno do gancho de pré-provisão

A função do Lambda deve retornar uma resposta que indique se autorizou a solicitação de provisionamento e os valores de quaisquer propriedades a serem substituídos.

Veja a seguir um exemplo de uma resposta bem-sucedida da função de pré-provisionamento.

```
{
    "allowProvisioning": true,
    "parameterOverrides" : {
        "Key": "newCustomValue",
        ...
    }
}
```

`"parameterOverrides"`Os valores de serão adicionados ao`"parameters"`parâmetro dothe section called “RegisterThing” (p. 833)Carga da solicitação.

Note

- Se a função do Lambda falhar, a solicitação de provisionamento falhará com`ACCESS_DENIED`e um erro é registrado em CloudWatch Logs.
- Se a função do Lambda não retornar`"allowProvisioning": "true"`na resposta, a solicitação de provisionamento falha com`ACCESS_DENIED`.
- A função do Lambda deve concluir a execução e retornar em até 5 segundos, caso contrário, a solicitação de provisionamento falhará.

Exemplo do Lambda do gancho de pré-provisionamento

Python

Um exemplo de um gancho de pré-provisionamento do Lambda no Python.

```
import json

def pre_provisioning_hook(event, context):
    print(event)

    return {
        'allowProvisioning': True,
        'parameterOverrides': {
            'DeviceLocation': 'Seattle'
        }
    }
```

Java

Um exemplo de um gancho de pré-provisionamento do Lambda em Java.

Classe do manipulador:

```
package example;

import java.util.Map;
import java.util.HashMap;
import com.amazonaws.services.lambda.runtime.Context;
import com.amazonaws.services.lambda.runtime.RequestHandler;

public class PreProvisioningHook implements RequestHandler<PreProvisioningHookRequest,
 PreProvisioningHookResponse> {

    public PreProvisioningHookResponse handleRequest(PreProvisioningHookRequest object,
 Context context) {
        Map<String, String> parameterOverrides = new HashMap<String, String>();
        parameterOverrides.put("DeviceLocation", "Seattle");

        PreProvisioningHookResponse response = PreProvisioningHookResponse.builder()
                .allowProvisioning(true)
                .parameterOverrides(parameterOverrides)
                .build();

        return response;
    }

}
```

Solicitar classe:

```
package example;

import java.util.Map;
import lombok.Builder;
import lombok.Data;
import lombok.AllArgsConstructor;
import lombok.NoArgsConstructor;

@Data
@Builder
@AllArgsConstructor
@NoArgsConstructor
public class PreProvisioningHookRequest {
    private String claimCertificateId;
    private String certificateId;
    private String certificatePem;
    private String templateArn;
    private String clientId;
    private Map<String, String> parameters;
}
```

Classe de resposta:

```
package example;

import java.util.Map;
import lombok.Builder;
import lombok.Data;
import lombok.AllArgsConstructor;
import lombok.NoArgsConstructor;
```

```
@Data
@Builder
@AllArgsConstructor
@NoArgsConstructor
public class PreProvisioningHookResponse {
    private boolean allowProvisioning;
    private Map<String, String> parameterOverrides;
}
```

API MQTT de provisionamento de dispositivos

O serviço de provisionamento de frotas oferece suporte a três APIs MQTT:

- the section called “CreateCertificateFromCsr” (p. 830)
- the section called “CreateKeysAndCertificate” (p. 832)
- the section called “RegisterThing” (p. 833)

Essa API oferece suporte a buffers de resposta no formato CBOR (Concise Binary Object Representation) e JavaScript Object Notation (JSON), dependendo do*formato da carga*do tópico. Por uma questão de clareza, no entanto, os exemplos de resposta e de solicitação nesta seção são mostrados no formato JSON.

formato da carga	Tipo de dados do formato de resposta
cbor	Concise Binary Object Representation (CBOR)
json	JavaScript Object Notation (JSON)

> **Important**
>
> Antes de publicar um tópico de mensagem de solicitação, assine os tópicos de resposta para receber a resposta. As mensagens usadas por esta API usam o protocolo MQTT de publicação/assinatura para fornecer uma interação de solicitação e resposta.
> Se você não assinar os tópicos de resposta antes de publicar uma solicitação, talvez não receba os resultados dessa solicitação.

CreateCertificateFromCsr

Cria um certificado a partir de uma solicitação de assinatura de certificado (CSR).AWS IoTO fornece certificados de cliente que são assinados pela autoridade de certificação (CA) raiz da Amazon. O novo certificado tem um status `PENDING_ACTIVATION`. Quando você chama `RegisterThing` para provisionar uma coisa com esse certificado, o status do certificado muda para `ACTIVE` ou `INACTIVE` conforme descrito no modelo.

> **Note**
>
> Por segurança, `ocertificateOwnershipToken`retornado por`CreateCertificateFromCsr` (p. 830)expira após uma hora.`RegisterThing` (p. 833)deve ser chamado antes do`certificateOwnershipToken`Expira. Se o certificado criado por`CreateCertificateFromCsr` (p. 830)não foi ativado e não foi anexado a uma política ou coisa no momento em que o token expirar, o certificado é excluído. Se o token expirar,

o dispositivo pode chamar`CreateCertificateFromCsr` (p. 830)Para gerar um novo certificado.

Solicitação CreateCertificateFromCsr

Publique uma mensagem com o tópico `$aws/certificates/create-from-csr/`*`payload-format`*.

`payload-format`

O formato da carga da mensagem como `cbor` ou `json`.

Carga da solicitação CreateCertificateFromCsr

```
{
    "certificateSigningRequest": "string"
}
```

`certificateSigningRequest`

A CSR, no formato PEM.

Resposta de CreateCertificateFromCsr

Assine `$aws/certificates/create-from-csr/`*`payload-format`*`/accepted`.

`payload-format`

O formato da carga da mensagem como `cbor` ou `json`.

Carga da resposta de CreateCertificateFromCsr

```
{
    "certificateOwnershipToken": "string",
    "certificateId": "string",
    "certificatePem": "string"
}
```

`certificateOwnershipToken`

O token para provar a propriedade do certificado durante o provisionamento.

`certificateId`

O ID do certificado. As operações de gerenciamento de certificado usam apenas um certificateId.

`certificatePem`

Os dados do certificado, no formato PEM.

Erro de CreateCertificateFromCsr

Para receber respostas de erro, assine `$aws/certificates/create-from-csr/`*`payload-format`*`/rejected`.

`payload-format`

O formato da carga da mensagem como `cbor` ou `json`.

Carga do erro de CreateCertificateFromCsr

```
{
    "statusCode": int,
    "errorCode": "string",
    "errorMessage": "string"
}
```

`statusCode`

O código do status.

`errorCode`

O código do erro.

`errorMessage`

A mensagem de erro.

CreateKeysAndCertificate

Cria chaves e um certificado.AWS IoTO fornece certificados de cliente que são assinados pela autoridade de certificação (CA) raiz da Amazon. O novo certificado tem um status `PENDING_ACTIVATION`. Quando você chama `RegisterThing` para provisionar uma coisa com esse certificado, o status do certificado muda para `ACTIVE` ou `INACTIVE` conforme descrito no modelo.

> **Note**
>
> Por segurança, o`certificateOwnershipToken`retornado por`CreateKeysAndCertificate` (p. 832)expira após uma hora.`RegisterThing` (p. 833)deve ser chamado antes do`certificateOwnershipToken`Expira. Se o certificado criado por`CreateKeysAndCertificate` (p. 832)não foi ativado e não foi anexado a uma política ou coisa no momento em que o token expirar, o certificado é excluído. Se o token expirar, o dispositivo pode chamar`CreateKeysAndCertificate` (p. 832)Para gerar um novo certificado.

Solicitação CreateKeysAndCertificate

Publique uma mensagem em `$aws/certificates/create/`*`payload-format`* com uma carga de mensagem vazia.

`payload-format`

O formato da carga da mensagem como `cbor` ou `json`.

Resposta de CreateKeysAndCertificate

Assine `$aws/certificates/create/`*`payload-format`*`/accepted`.

`payload-format`

O formato da carga da mensagem como `cbor` ou `json`.

Resposta de CreateKeysAndCertificate

```
{
```

```
    "certificateId": "string",
    "certificatePem": "string",
    "privateKey": "string",
    "certificateOwnershipToken": "string"
}
```

`certificateId`

O ID do certificado.

`certificatePem`

Os dados do certificado, no formato PEM.

`privateKey`

A chave privada.

`certificateOwnershipToken`

O token para provar a propriedade do certificado durante o provisionamento.

Erro de CreateKeysAndCertificate

Para receber respostas de erro, assine `$aws/certificates/create/`*`payload-format`*`/rejected`.

`payload-format`

O formato da carga da mensagem como `cbor` ou `json`.

Carga do erro de CreateKeysAndCertificate

```
{
    "statusCode": int,
    "errorCode": "string",
    "errorMessage": "string"
}
```

`statusCode`

O código do status.

`errorCode`

O código do erro.

`errorMessage`

A mensagem de erro.

RegisterThing

Provisiona uma coisa usando um modelo predefinido.

Solicitação RegisterThing

Publique uma mensagem em `$aws/provisioning-templates/`*`templateName`*`/provision/`*`payload-format`*.

`payload-format`

O formato da carga da mensagem como `cbor` ou `json`.

`templateName`

O nome do modelo provisionado.

Carga da solicitação RegisterThing

```
{
    "certificateOwnershipToken": "string",
    "parameters": {
        "string": "string",
        ...
    }
}
```

`certificateOwnershipToken`

O token para provar a propriedade do certificado. O token é gerado pela AWS IoT quando você cria um certificado por MQTT.

`parameters`

Opcional. Pares de chave-valor do dispositivo que são usados pelos ganchos de pré-provisionamento (p. 827) para avaliar a solicitação de registro.

Resposta de RegisterThing

Assine `$aws/provisioning-templates/`*`templateName`*`/provision/`*`payload-format`*`/accepted`.

`payload-format`

O formato da carga da mensagem como `cbor` ou `json`.

`templateName`

O nome do modelo provisionado.

Carga da resposta de RegisterThing

```
{
    "deviceConfiguration": {
        "string": "string",
        ...
    },
    "thingName": "string"
}
```

`deviceConfiguration`

A configuração do dispositivo definida no modelo.

`thingName`

O nome da coisa da IoT criada durante o provisionamento.

Resposta do erro de RegisterThing

Para receber respostas de erro, assine `$aws/provisioning-templates/`*`templateName`*`/provision/`*`payload-format`*`/rejected`.

`payload-format`

O formato da carga da mensagem como `cbor` ou `json`.

`templateName`

O nome do modelo provisionado.

Carga da resposta ao erro de RegisterThing

```
{
    "statusCode": int,
    "errorCode": "string",
    "errorMessage": "string"
}
```

`statusCode`

O código do status.

`errorCode`

O código do erro.

`errorMessage`

A mensagem de erro.

Indexação de frota

> **Note**
>
> O recurso de indexação de frota para suportar indexação denominada sombras eAWS IoT Device DefenderOs dados de violações estão em versão de visualização paraAWS IoT Device ManagementE está sujeita a alterações.

Você pode usar a indexação de frota para indexar, pesquisar e agregar os dados de seus dispositivos das seguintes fontes:AWS IoTregistro (p. 265),AWS IoTDevice Shadow (p. 633),AWS IoTconectividade (p. 1121), eAWS IoT Device Defender (p. 882)Violações. Você pode consultar um grupo de dispositivos e agregar estatísticas em registros de dispositivos baseados em diferentes combinações de atributos de dispositivo, incluindo estado, conectividade e violações de dispositivo. Com a indexação de frotas, você pode organizar, investigar e solucionar problemas de sua frota de dispositivos.

A indexação de frota fornece os seguintes recursos.

- Gerenciar atualizações de índice

Depois de configurar um índice de frota para que ele indexe atualizações de seus grupos, registros, sombra de dispositivo e violações de dispositivo. Quando você habilita a indexação da frota,AWS IoTO cria um índice para suas coisas ou grupos de coisas.`AWS_Things`é o índice criado para todas as suas coisas.`AWS_ThingGroups`é o índice que contém todos os seus grupos de coisas. Depois que a indexação de frota estiver ativa, você pode executar consultas no índice, por exemplo, para encontrar todos os dispositivos portáteis e ter uma duração da bateria maior que 70%.AWS IoTMantém o índice continuamente atualizado com os dados mais recentes. Para obter mais informações, consulteGerenciamento de indexação de frotas (p. 837).

- Pesquisando em fontes de dados

Você pode criar uma string de consulta com base emUma linguagem de consulta simples (p. 857)e use-o para pesquisar nas fontes de dados que você configura na configuração de indexação da frota. A string de consulta descreve as coisas que você deseja encontrar. Para obter mais informações sobre fontes de dados que são compatíveis com a indexação de frota, consulteGerenciar a indexação de coisas (p. 840).

- Consulta de dados agregados

Você pode pesquisar em seus dispositivos dados agregados e estatísticas de retorno, percentil, cardinalidade ou uma lista de coisas com consultas de pesquisa referentes a campos específicos. Para obter mais informações sobre consulta de agregação, consulteConsulta de dados agregados (p. 851).

- Monitorando dados agregados usando métricas de frota

Você pode usar métricas de frota para enviar automaticamente dados agregados ao CloudWatch, analisando tendências e criando alarmes para monitorar o estado agregado de sua frota. Para obter mais informações sobre métricas de frota, consulteMétricas de frota (p. 862).

Para usar a indexação de frota, você deve configurar a configuração de indexação da frota. Para configurar a configuração de indexação de frota, você pode usar oAWS IoTconsole, ou se preferir acesso programático, poderá usar aAWSSDKs ou oAWS Command Line Interface(AWS CLI).

Para obter informações sobre a definição de preço deste e de outros serviços, consulteAWS IoTPreços do Gerenciamento de dispositivos do.

Gerenciamento de indexação de frotas

> **Note**
>
> O recurso de indexação de frota para suportar indexação denominada sombras eAWS IoT Device DefenderOs dados de violações estão em versão de visualização paraAWS IoT Device Managemente sujeita a alterações.

A indexação de frota gerencia dois tipos de índices para você, indexação de coisas e indexação de grupos de coisas.

Indexação de coisas

O índice criado para todas as suas coisas é`AWS_Things`. A indexação de coisas dá suporte às seguintes fontes de dados:AWS IoTregistro (p. 265)dados,AWS IoTDevice Shadow (p. 633)dados,AWS IoTconectividade (p. 1121)dados, eAWS IoT Device Defender (p. 882)dados de violações.

Registro do-AWS IoTA fornece um registro de coisas que ajuda você a gerenciar as coisas. Com a indexação de frota, você pode adicionar coisas ao registro e pesquisar por dispositivos. Para obter mais informações sobre o registro, consulteComo gerenciar coisas com o registro (p. 265).

Shadow-OAWS IoTServiço Device Shadow do (p. 633)fornece sombras que ajudam você a armazenar os dados de estado de seus dispositivos. A indexação de coisas suporta sombras clássicas sem nome e sombras nomeadas. Para obter mais informações sobre sombras, consulteAWS IoTServiço Device Shadow do (p. 633).

Conectividade-Os dados de conectividade do dispositivo ajudam a identificar o status da conexão de seus dispositivos. Esses dados de conectividade são orientados porEventos de ciclo de vida (p. 1121). Quando um cliente se conecta ou se desconecta,AWS IoTPublica eventos do ciclo de vida com mensagens em tópicos do MQTT. Uma mensagem de conexão ou desconexão pode ser uma lista de elementos JSON que fornecem detalhes do status da conexão. Para obter mais informações sobre conectividade de dispositivos, consulteEventos de ciclo de vida (p. 1121).

Violações do Device Def-AWS IoT Device Defenderos dados de violações ajudam a identificar os comportamentos anômalos de seus dispositivos em relação aos comportamentos normais definidos em um Perfil de segurança. Um perfil de segurança contém um conjunto de comportamentos de dispositivo esperados. Cada comportamento usa uma métrica que especifica o comportamento normal de seus dispositivos. Para obter mais informações sobre violações do Device Defender, consulteAWS IoT Device DefenderDetectar (p. 993).

Para obter mais informações, consulteGerenciar a indexação de coisas (p. 840).

Indexação de grupos de coisas

`AWS_ThingGroups` é o índice que contém todos os seus grupos de coisas. Você pode usar esse índice para pesquisar grupos com base no nome do grupo, na descrição, nos atributos e em todos os nomes de grupo pai.

Para obter mais informações, consulteGerenciar a indexação de grupos de coisas (p. 849).

Campos gerenciados

Os campos gerenciados contêm dados associados a coisas, grupos de coisas, sombras de dispositivo, conectividade de dispositivo e violações do Device Defender.AWS IoTdefine o tipo de dados em campos gerenciados. Especifique os valores de cada campo gerenciado ao criar uma coisa da IoT. Por exemplo, nomes de coisas, grupos de coisas e descrições de coisas são todos os campos gerenciados. Os índices de indexação de frota são gerenciados com base no modo de indexação especificado. Os campos gerenciados não podem ser alterados nem aparecem em`customFields`. Para obter mais informações, consulteCampos personalizados (p. 839).

A seguir lista campos gerenciados para indexação de coisas:

- Campos gerenciados para o registro

```
"managedFields" : [
  {name:thingId, type:String},
  {name:thingName, type:String},
  {name:registry.version, type:Number},
  {name:registry.thingTypeName, type:String},
  {name:registry.thingGroupNames, type:String},
]
```

- Campos gerenciados para sombras clássicas sem nome

```
"managedFields" : [
  {name:shadow.version, type:Number},
  {name:shadow.hasDelta, type:Boolean}
]
```

- Campos gerenciados para sombras nomeadas

```
"managedFields" : [
  {name:shadow.name.shadowName.version, type:Number},
  {name:shadow.name.shadowName.hasDelta, type:Boolean}
]
```

- Campos gerenciados para conectividade de coisas

```
"managedFields" : [
  {name:connectivity.timestamp, type:Number},
  {name:connectivity.version, type:Number},
  {name:connectivity.connected, type:Boolean},
  {name:connectivity.disconnectReason, type:String}
]
```

- Campos gerenciados para Device Defender

```
"managedFields" : [
  {name:deviceDefender.violationCount, type:Number},
  {name:deviceDefender.securityprofile.behaviorname.metricName, type:String},
  {name:deviceDefender.securityprofile.behaviorname.lastViolationTime, type:Number},
  {name:deviceDefender.securityprofile.behaviorname.lastViolationValue, type:String},
  {name:deviceDefender.securityprofile.behaviorname.inViolation, type:Boolean}
]
```

- Campos gerenciados para grupos de coisas

```
"managedFields" : [
  {name:description, type:String},
  {name:parentGroupNames, type:String},
```

```
  {name:thingGroupId, type:String},
  {name:thingGroupName, type:String},
  {name:version, type:Number},
]
```

A tabela a seguir lista os campos gerenciados que não são pesquisáveis.

Fonte de dados	Campo gerenciado que não é pesquisável
Registro	`registry.version`
Sombras sem nome	`shadow.version`
Sombras nomeadas	`shadow.name.*.version`
Device Defender	`deviceDefender.version`
Grupos de coisas	`version`

Campos personalizados

Você pode agregar atributos de coisas, dados do Device Shadow e dados de violações do Device Defender criando campos personalizados para indexá-los. O`customFields`O atributo é uma lista de pares de nome de campo e tipos de dados. Você pode executar consultas de agregação com base no tipo de dados. O modo de indexação escolhido afeta quais campos podem ser especificados em`customFields`. Por exemplo, se você especificar o`REGISTRY`O modo de indexação, não é possível especificar um campo personalizado a partir de uma sombra de coisa. Você pode usar oatualização de indexação de configuraçãoComando da CLI para criar ou atualizar os campos personalizados (consulte um comando de exemplo emAtualizando exemplos de configuração de indexação (p. 842)).

- Nomes de campos personalizados

Nomes de campos personalizados para atributos de grupo de coisas e coisas começam com`attributes.`, seguido pelo nome do atributo. Se a indexação de sombra sem nome estiver ativada, as coisas podem ter nomes de campos personalizados que começam com`shadow.desired`ou`shadow.reported`, seguido pelo nome do valor de dados de sombra sem nome. Se a indexação de sombra nomeada estiver ativada, as coisas podem ter nomes de campos personalizados que começam com`shadow.name.*.desired.`ou`shadow.name.*.reported.`, seguido pelo valor de dados de sombra nomeado. Se a indexação de violações do Device Defender estiver ativada, as coisas poderão ter nomes de campos personalizados que começam com`deviceDefender.`, seguido pelo valor de dados de violações do Device Defender.

O nome do atributo ou do valor de dados que segue o prefixo pode ter apenas caracteres alfanuméricos, - (hífen) e _ (sublinhado). Ele nome não pode ter espaços.

Se houver uma inconsistência de tipo entre um campo personalizado na configuração e o valor que está sendo indexado, a indexação de frota ignorará o valor inconsistente para consultas de agregação. Os logs do CloudWatch são úteis ao solucionar problemas de consulta de agregação. Para obter mais informações, consulte Solução de problemas de consultas de agregação para o serviço de indexação de frota (p. 1288).

- Tipos de campos personalizados

Os tipos de campo personalizados têm os seguintes valores compatíveis:`Number`,`String`, e`Boolean`.

Gerenciar a indexação de coisas

> **Note**
>
> O recurso de indexação de frota para suportar indexação denominada sombras eAWS IoT Device DefenderOs dados de violações estão em versão de visualização paraAWS IoT Device Managemente sujeita a alterações.

O índice criado para todas as suas coisas é`AWS_Things`. Você pode controlar o que indexar das seguintes fontes de dados:AWS IoTregistro (p. 265)dados,AWS IoTDevice Shadow (p. 633)dados,AWS IoTconectividade (p. 1121)dados, eAWS IoT Device Defender (p. 882)dados de violações.

Habilitar a indexação de coisas

Você usa oatualização de indexação de configuraçãoO comando da CLI ou oUpdateIndexingConfigurationOperação de API para criar o`AWS_Things`indexe e controle sua configuração. Ao usar o`--thing-indexing-configuration`(`thingIndexingConfiguration`), você controla que tipo de dados (por exemplo, dados de registro, sombra, conectividade de dispositivo e dados de violações do Device Defender) está indexado.

O parâmetro `--thing-indexing-configuration` leva uma string com a seguinte estrutura:

```
{
  "thingIndexingMode": "OFF"|"REGISTRY"|"REGISTRY_AND_SHADOW",
  "thingConnectivityIndexingMode": "OFF"|"STATUS",
  "deviceDefenderIndexingMode": "OFF"|"VIOLATIONS",
  "namedShadowIndexingMode": "OFF"|"ON",
  "managedFields": [
    {
      "name": "string",
      "type": "Number"|"String"|"Boolean"
    },
    ...
  ],
  "customFields": [
    {
      "name": "string",
      "type": "Number"|"String"|"Boolean"
    },
    ...
  ]
}
```

Modo de indexação de coisas

O atributo `thingIndexingMode` controla o tipo de dados indexados.

> **Important**
>
> Para habilitar a indexação de coisas, o`thingIndexingMode`atributo não pode ser definido como DESLIGADO.

Atributo	Valores válidos	Descrição
`thingIndexingMode`	DESL.	Sem indexação.

Atributo	Valores válidos	Descrição
	REGISTRY	Dados de registro de índice.
	REGISTRY_AND_SHADOW	Registro de índice e dados de sombra de coisa.

O atributo `thingConnectivityIndexingMode` especifica se dados de conectividade são indexados.

Atributo	Valores válidos	Descrição
`thingConnectivityIndexingMode`	Não especificado.	A coisa que os dados de conectividade não são indexados.
	DESL.	A coisa que os dados de conectividade não são indexados.
	STATUS	A coisa que os dados de conectividade são indexados.

O`deviceDefenderIndexingMode`O atributo especifica se dados de violações do Device Defender são indexados.

Atributo	Valores válidos	Descrição
`deviceDefenderIndexingMode`	Não especificado.	Os dados de violações do Device Defender não são indexados.
	DESL.	Os dados de violações do Device Defender não são indexados.
	VIOLAÇÕES	Os dados de violações do Device Defender são indexados.

O`namedShadowIndexingMode`O atributo especifica se dados de sombra nomeados são indexados.

Atributo	Valores válidos	Descrição
`namedShadowIndexingMode`	Não especificado.	Dados de sombra nomeados não são indexados.
	DESL.	Dados de sombra nomeados não são indexados.
	ON	Dados de sombra nomeados são indexados.

Campos gerenciados e campos personalizados

Campos gerenciados

Os campos gerenciados contêm dados associados a coisas, grupos de coisas, sombras de dispositivo, conectividade de dispositivo e violações do Device Defender.AWS IoTdefine o tipo de dados em campos gerenciados. Especifique os valores de cada campo gerenciado ao criar uma coisa da IoT. Por exemplo, nomes de coisas, grupos de coisas e descrições de coisas são todos os campos gerenciados. Os índices de indexação de frota são gerenciados com base no modo de indexação especificado. Os campos gerenciados não podem ser alterados nem aparecem em`customFields`.

Campos personalizados

Você pode agregar atributos, dados do Device Shadow e dados de violações do Device Defender criando campos personalizados para indexá-los. O`customFields`O atributo é uma lista de pares de nome de campo e tipos de dados. Você pode executar consultas de agregação com base no tipo de dados. O modo de indexação escolhido afeta quais campos podem ser especificados em`customFields`. Por exemplo, se você especificar o`REGISTRY`O modo de indexação, não é possível especificar um campo personalizado a partir de uma sombra de coisa. Você pode usar oatualização de indexação de configuraçãoComando da CLI para criar ou atualizar os campos personalizados (consulte um comando de exemplo emAtualizando exemplos de configuração de indexação (p. 842)). Para obter mais informações, consulteCampos personalizados (p. 839).

Atualizando exemplos de configuração de indexação

Você pode usar o comando AWS IoT update-indexing-configuration da CLI para atualizar a configuração da indexação. Os exemplos a seguir mostram como usarupdate-indexing-configuration.

Sintaxe curta:

```
aws iot update-indexing-configuration --thing-indexing-configuration \
'thingIndexingMode=REGISTRY_AND_SHADOW,deviceDefenderIndexingMode=VIOLATIONS,namedShadowIndexingMode=ON
{name= shadow.name.thing1shadow.desired.DefaultDesired,
 type=String},{name=shadow.desired.power, type=Boolean},
 {name=deviceDefender.securityProfile1.NUMBER_VALUE_BEHAVIOR.lastViolationValue.number,
 type=Number}]'
```

Sintaxe do JSON:

```
aws iot update-indexing-configuration --cli-input-json \ '{
          "thingIndexingConfiguration": { "thingIndexingMode": "REGISTRY_AND_SHADOW",
          "thingConnectivityIndexingMode": "STATUS",
          "deviceDefenderIndexingMode": "VIOLATIONS",
          "namedShadowIndexingMode": "ON",
          "customFields": [ { "name": "shadow.desired.power", "type": "Boolean" },
          {"name": "attributes.version", "type": "Number"},
          {"name": "shadow.name.thing1shadow.desired.DefaultDesired", "type": "String"},
          {"name":
 "deviceDefender.securityProfile1.NUMBER_VALUE_BEHAVIOR.lastViolationValue.number", "type":
 Number} ] } }
```

Esse comando não produz nenhuma saída.

Para verificar o status do índice da coisa, execute o`describe-index`Comando da CLI:

```
aws iot describe-index --index-name "AWS_Things"
```

A saída do comando `describe-index` é semelhante a:

```
{
    "indexName": "AWS_Things",
    "indexStatus": "ACTIVE",
    "schema": "MULTI_INDEXING_MODE"
```

```
}
```

> **Note**
>
> Pode levar um momento para a indexação da frota atualizar o índice da frota. Recomendamos aguardar até `oindexStatus`mostra ACTIVE antes de usá-lo. Você pode ter valores diferentes no campo esquema, dependendo de quais fontes de dados você configurou. Para obter mais informações, consulteDescrever um índice de coisas (p. 845).

Para obter os detalhes da configuração de indexação, execute `oget-indexing-configuration`Comando da CLI:

```
aws iot get-indexing-configuration
```

A saída do comando `get-indexing-configuration` é semelhante a:

```
{
    "thingIndexingConfiguration": {
        "thingIndexingMode": "REGISTRY_AND_SHADOW",
        "thingConnectivityIndexingMode": "STATUS",
        "deviceDefenderIndexingMode": "VIOLATIONS",
        "namedShadowIndexingMode": "ON",
        "managedFields": [
            {
                "name": "connectivity.disconnectReason",
                "type": "String"
            },
            {
                "name": "registry.version",
                "type": "Number"
            },
            {
                "name": "thingName",
                "type": "String"
            },
            {
                "name": "deviceDefender.violationCount",
                "type": "Number"
            },
            {
                "name": "shadow.hasDelta",
                "type": "Boolean"
            },
            {
                "name": "shadow.name.*.version",
                "type": "Number"
            },
            {
                "name": "shadow.version",
                "type": "Number"
            },
            {
                "name": "connectivity.version",
                "type": "Number"
            },
            {
                "name": "connectivity.timestamp",
                "type": "Number"
            },
            {
                "name": "shadow.name.*.hasDelta",
                "type": "Boolean"
            },
```

```
            {
                "name": "registry.thingTypeName",
                "type": "String"
            },
            {
                "name": "thingId",
                "type": "String"
            },
            {
                "name": "connectivity.connected",
                "type": "Boolean"
            },
            {
                "name": "registry.thingGroupNames",
                "type": "String"
            }
        ],
        "customFields": [
            {
                "name": "shadow.name.thing1shadow.desired.DefaultDesired",
                "type": "String"
            },

            {
                "name":
 "deviceDefender.securityProfile1.NUMBER_VALUE_BEHAVIOR.lastViolationValue.number",
                "type": "Number"
            },
            {
                "name": "shadow.desired.power",
                "type": "Boolean"
            },
            {
                "name": "attributes.version",
                "type": "Number"
            }
        ]
    },
    "thingGroupIndexingConfiguration": {
        "thingGroupIndexingMode": "OFF"
    }
}
```

Para atualizar os campos personalizados, você pode executar o comando update-indexing-configuration. Um exemplo é o seguinte:

```
aws iot update-indexing-configuration --thing-indexing-configuration

 'thingIndexingMode=REGISTRY_AND_SHADOW,customFields=[{name=attributes.version,type=Number},
{name=attributes.color,type=String},{name=shadow.desired.power,type=Boolean},
{name=shadow.desired.intensity,type=Number}]'
```

Esse comando adicionou `shadow.desired.intensity` à configuração de indexação.

> **Note**
>
> Atualizar a configuração de indexação de campo personalizado substitui todos os campos personalizados existentes. Especifique todos os campos personalizados ao chamar update-indexing-configuration.

Depois que o índice é reconstruído, você pode usar uma consulta de agregação nos campos recém-adicionados, dados de registro de pesquisa, dados de sombra e dados de status de conectividade de coisa.

Ao alterar o modo de indexação, verifique se todos os campos personalizados são válidos usando o novo modo de indexação. Por exemplo, se você começar a usar`REGISTRY_AND_SHADOW`modo com um campo personalizado chamado`shadow.desired.temperature`, você deve excluir o`shadow.desired.temperature`campo personalizado antes de alterar o modo de indexação para`REGISTRY`. Se a configuração de indexação contiver campos personalizados que não são indexados pelo modo de indexação, a atualização falhará.

Descrever um índice de coisas

O comando a seguir mostra como usar o comando describe-index da CLI para recuperar o status atual do índice de coisas.

```
aws iot describe-index --index-name "AWS_Things"
```

A resposta do comando pode ser parecida com o seguinte:

```
{
    "indexName": "AWS_Things",
    "indexStatus": "BUILDING",
    "schema": "REGISTRY_AND_SHADOW_AND_CONNECTIVITY_STATUS"
}
```

Na primeira vez que você habilita a indexação de frota,AWS IoTconstrói seu índice. Quando`indexStatus`está no`BUILDING`State, você não pode consultar o índice. O `schema` do índice de coisas indica qual tipo de dados (`REGISTRY_AND_SHADOW_AND_CONNECTIVITY_STATUS`) é indexado.

A alteração da configuração do índice faz com que o índice seja recompilado. Durante esse processo, o `indexStatus` é `REBUILDING`. Você pode executar consultas em dados no índice de coisas enquanto ele está sendo reconstruído. Por exemplo, se você alterar a configuração do índice de `REGISTRY` para `REGISTRY_AND_SHADOW` enquanto o índice estiver sendo recompilado, poderá consultar dados do registro, incluindo as últimas atualizações. No entanto, você não pode consultar os dados de shadow até que a reconstrução seja concluída. O tempo necessário para compilar ou recompilar o índice depende da quantidade de dados.

Você pode ver valores diferentes no campo de esquema, dependendo das fontes de dados que você configurou. A tabela abaixo mostra os diferentes valores de esquema e as descrições correspondentes:

Esquema	Descrição
DESL.	Nenhuma fonte de dados é configurada ou indexada.
REGISTRY	Os dados do registro são indexados.
REGISTRY_AND_SHADOW	Os dados do registro e dados de shadow sem nome (clássicos) são indexados.
REGISTRY_AND_CONNECTIVITY	Dados de registro e dados de conectividade são indexados.
REGISTRY_AND_SHADOW_AND_CONNECTIVITY_STATUS	Dados de registro, dados de sombra sem nome (clássicos) e dados de conectividade são indexados.
MULTI_INDEXING_MODE	Os dados de violações de sombra ou Device Defender nomeados são indexados, além do registro, sombra sem nome (clássico) ou dados de conectividade.

Consultar um índice de coisas

Use o comando search-index da CLI para consultar dados no índice.

```
aws iot search-index --index-name "AWS_Things" --query-string
          "thingName:mything*"
```

```
{
    "things":[{
         "thingName":"mything1",
         "thingGroupNames":[
            "mygroup1"
         ],
         "thingId":"a4b9f759-b0f2-4857-8a4b-967745ed9f4e",
         "attributes":{
            "attribute1":"abc"
         },
         "connectivity": {
            "connected":false,
            "timestamp":1556649874716,
            "disconnectReason": "CONNECTION_LOST"
         }
    },
    {
        "thingName":"mything2",
        "thingTypeName":"MyThingType",
        "thingGroupNames":[
            "mygroup1",
            "mygroup2"
        ],
        "thingId":"01014ef9-e97e-44c6-985a-d0b06924f2af",
        "attributes":{
            "model":"1.2",
            "country":"usa"
        },
        "shadow":{
            "desired":{
                "location":"new york",
                "myvalues":[3, 4, 5]
            },
            "reported":{
                "location":"new york",
                "myvalues":[1, 2, 3],
                "stats":{
                    "battery":78
                }
            },
            "metadata":{
                 "desired":{
                       "location":{
                            "timestamp":123456789
                        },
                       "myvalues":{
                             "timestamp":123456789
                       }
                  },
                  "reported":{
                        "location":{
                             "timestamp":34535454
                         },
                        "myvalues":{
                             "timestamp":34535454
                        },
```

```
                        "stats":{
                             "battery":{
                                   "timestamp":34535454
                             }
                        }
                   }
              },
              "version":10,
              "timestamp":34535454
         },
         "connectivity": {
              "connected":true,
              "timestamp":1556649855046
         }
    }],
    "nextToken":"AQFCuvk7zZ3D9pOYMbFCeHbdZ+h=G"
}
```

Na resposta JSON,`"connectivity"`(conforme habilitado pelo`thingConnectivityIndexingMode=STATUS`setting) fornece um valor booliano, um registro de data e hora e um DisconnectReason que indica se o dispositivo está conectado aoAWS IoT Core. O dispositivo`"mything1"`desconectado (`false`) no horário POSIX`1556649874716`devido a`CONNECTION_LOST`. Para obter mais informações sobre motivos de desconexão, consulteEventos de ciclo de vida (p. 1121).

```
"connectivity": {
    "connected":false,
    "timestamp":1556649874716,
    "disconnectReason": "CONNECTION_LOST"
}
```

O dispositivo `"mything2"` conectado (`true`) no horário POSIX `1556649855046`:

```
"connectivity": {
    "connected":true,
    "timestamp":1556649855046
}
```

Os carimbos de data e hora são dados em milissegundos desde a época, então`1556649855046`Representa 18h44min15seg046 na terça-feira, 30 de abril de 2019 (UTC).

> **Important**
>
> Se um dispositivo foi desconectado por aproximadamente uma hora, o`"timestamp"`valor e o`"disconnectReason"`O valor do status de conectividade pode estar ausente.

Restrições e limitações

Estas são as restrições e limitações para `AWS_Things`.

Campos de shadow com tipos complexos

Um campo de shadow é indexado somente se o valor do campo é um tipo simples, por exemplo, um objeto JSON que não contém uma matriz ou uma matriz que consiste inteiramente em tipos simples. Tipo simples significa uma string, um número ou um dos literais `true` ou `false`. Por exemplo, tendo em vista o seguinte estado do shadow, o valor do campo`"palette"`O não é indexado porque é uma matriz que contém itens de tipos complexos. O valor de campo `"colors"` é indexado, pois cada valor na matriz é uma string.

```
{
    "state": {
```

```
        "reported": {
            "switched": "ON",
            "colors": [ "RED", "GREEN", "BLUE" ],
            "palette": [
                {
                    "name": "RED",
                    "intensity": 124
                },
                {
                    "name": "GREEN",
                    "intensity": 68
                },
                {
                    "name": "BLUE",
                    "intensity": 201
                }
            ]
        }
    }
}
```

Nomes de campo de shadow aninhados

Os nomes de campos de shadow aninhados são armazenados como uma string delimitada por ponto (.). Por exemplo, considerando um documento de shadow:

```
{
  "state": {
    "desired": {
      "one": {
        "two": {
          "three": "v2"
        }
      }
    }
  }
}
```

O nome do campo`three`É armazenado como`desired.one.two.three`. Se você também tiver um documento de shadow, ele será armazenado assim:

```
{
  "state": {
    "desired": {
      "one.two.three": "v2"
    }
  }
}
```

ambos correspondem a uma consulta para`shadow.desired.one.two.three:v2`. Como prática recomendada, não use pontos em nomes de campo de shadow.

Metadados de shadow

Um campo de uma seção de metadados de shadow é indexado, mas somente se o campo correspondente na seção `"state"` do shadow for indexado. (No exemplo anterior, o`"palette"`O campo na seção de metadados do shadow também não está indexado.)

Shadows não registrados

Se você usarUpdateThingShadowPara criar uma sombra usando um nome de coisa que não foi registrado noAWS IoTconta, campos nesta sombra não são indexados. Isso se aplica tanto à sombra clássica sem nome quanto à sombra nomeada.

Valores numéricos

Se qualquer registro ou dado de shadow for reconhecido pelo serviço como um valor numérico, ele será indexado dessa maneira. Você pode realizar consultas envolvendo intervalos e operadores de comparação de valores numéricos, (por exemplo `"attribute.foo<5"` ou `"shadow.reported.foo:[75 TO 80]"`). Para ser reconhecido como numérico, o valor dos dados deve ser um número JSON válido e literal. O valor pode ser um inteiro no intervalo -2 ^ 53... 2 ^ 53-1, um ponto flutuante de precisão dupla com opcional notação exponencial ou parte de uma matriz que contém somente valores como esse.

Valores nulos

Valores nulos não são indexados.

Valores máximos

O número máximo de campos personalizados para consultas de agregação é 5.

O número máximo de percentis solicitados para consultas de agregação é 100.

O tamanho total máximo de dados para uma coisa processada pela indexação da frota é limitado a 32 KB. Esses dados incluem dados indexados do registro, sombras clássicas e nomeadas, eventos do ciclo de vida de conectividade e dados de violações do Device Defender.

O número máximo de sombras nomeadas por coisa é 5.

A largura de banda máxima suportada pela indexação de frota é de 32 MBps.

Autorização

Você pode especificar o índice de coisas como um Nome de recurso da Amazon (ARN) em umAWS IoTação política, da seguinte forma.

Ação	Recurso
`iot:SearchIndex`	O ARN de um índice (por exemplo, `arn:aws:iot:`*`your-aws-regionyour-aws-account`*`:index/AWS_Things`).
`iot:DescribeIndex`	O ARN de um índice (por exemplo, `arn:aws:iot:`*`your-aws-region`*`:index/AWS_Things`).

Note

Se você tiver permissões para consultar o índice de frota, você poderá acessar os dados das coisas em toda a frota.

Gerenciar a indexação de grupos de coisas

`AWS_ThingGroups` é o índice que contém todos os seus grupos de coisas. Você pode usar esse índice para pesquisar grupos com base no nome do grupo, na descrição, nos atributos e em todos os nomes de grupo pai.

Habilitar a indexação de grupos de coisas

Você pode usar a configuração `thing-group-indexing-configuration` na API UpdateIndexingConfiguration para criar o índice `AWS_ThingGroups` e controlar a configuração. Você pode usar a API GetIndexingConfiguration para recuperar a configuração da indexação atual.

Para atualizar as configurações de indexação de grupos de coisas, execute oupdate-indexing-configurationComando da CLI:

```
aws iot update-indexing-configuration --thing-group-indexing-configuration
 thingGroupIndexingMode=ON
```

Você também pode atualizar configurações da indexação de coisas e grupos de coisas em um único comando, da seguinte forma:

```
aws iot update-indexing-configuration --thing-indexing-configuration
 thingIndexingMode=REGISTRY --thing-group-indexing-configuration thingGroupIndexingMode=ON
```

Estes são valores válidos para `thingGroupIndexingMode`.

DESL.

Sem indexação/excluir índice.

ON

Criar ou configurar o índice `AWS_ThingGroups`.

Para recuperar as configurações atuais de coisas e grupos de coisas, execute oget-indexing-configurationComando da CLI:

```
aws iot get-indexing-configuration
```

A resposta do comando é semelhante ao seguinte:

```
{
   "thingGroupIndexingConfiguration": {
        "thingGroupIndexingMode": "ON"
    }
}
```

Descrever índices de grupos

Para recuperar o status atual do`AWS_ThingGroups`index, use odescribe-indexComando da CLI:

```
aws iot describe-index --index-name "AWS_ThingGroups"
```

A resposta do comando é semelhante ao seguinte:

```
{
   "indexStatus": "ACTIVE",
   "indexName": "AWS_ThingGroups",
   "schema": "THING_GROUPS"
}
```

Na primeira vez que você habilita a indexação, a AWS IoT cria o índice. Você não pode consultar o índice se o `indexStatus` é `BUILDING`.

Consultar um índice de grupos de coisas

Para consultar dados no índice, use asearch-indexComando da CLI:

```
aws iot search-index --index-name "AWS_ThingGroups" --query-string
 "thingGroupName:mythinggroup*"
```

Autorização

Você pode especificar o índice de grupo de coisas como um ARN do recurso em uma ação de política da AWS IoT, da seguinte forma.

Ação	Recurso
`iot:SearchIndex`	O ARN de um índice (por exemplo, `arn:aws:iot:`*`your-aws-region`*`:index/AWS_ThingGroups`).
`iot:DescribeIndex`	O ARN de um índice (por exemplo, `arn:aws:iot:`*`your-aws-region`*`:index/AWS_ThingGroups`).

Consulta de dados agregados

AWS IoTfornece quatro APIs (`GetStatistics`,`GetCardinality`,`GetPercentiles`, e`GetBucketsAggregation`) que permitem que você pesquise dados agregados na frota de dispositivos.

> **Note**
>
> Para problemas com valores ausentes ou inesperados para as APIs de agregação, leiaGuia de solução de problemas de indexação de frota (p. 1288).

GetStatistics

A API GetStatistics e o comando get-statistics da CLI retornam a contagem, a média, a soma, o mínimo, o máximo, a soma de quadrados, a variância e o desvio padrão para o campo agregado especificado.

O comando da CLI get-statistics usa os seguintes parâmetros:

`index-name`

O nome do índice a ser pesquisado. O valor padrão é `AWS_Things`.

`query-string`

A consulta usada para pesquisar o índice. É possível especificar`"*"`Para obter a contagem de todas as coisas indexadas em seuConta da AWS.

`aggregationField`

Opcional. O campo a ser agregado. Esse campo deve ser um campo gerenciado ou personalizado definido ao chamar update-indexing-configuration. Se você não especificar um campo de agregação, `registry.version` será usado como campo de agregação.

`query-version`

A versão da consulta a ser usada. O valor padrão é `2017-09-30`.

O tipo de campo de agregação pode afetar as estatísticas retornadas.

GetStatistics com valores de string

Se você agregar em um campo de string, chamar `GetStatistics` retornará uma contagem de dispositivos que têm atributos que correspondem à consulta. Por exemplo:

```
aws iot get-statistics --aggregation-field 'attributes.stringAttribute'
            --query-string '*'
```

Esse comando retorna o número de dispositivos que contêm um atributo chamado `stringAttribute`:

```
{
  "statistics": {
    "count": 3
  }
}
```

GetStatistics com valores boolianos

Quando você chama `GetStatistics` com um campo de agregação booleano:

- AVERAGE é a porcentagem de dispositivos que correspondem à consulta.
- MINIMUM é 0 ou 1 de acordo com as seguintes regras:
  - Se todos os valores do campo de agregação forem `false`, MINIMUM será 0.
  - Se todos os valores do campo de agregação forem `true`, MINIMUM será 1.
  - Se os valores do campo de agregação forem uma mistura de `false` e `true`, MINIMUM será 0.
- MAXIMUM é 0 ou 1 de acordo com as seguintes regras:
  - Se todos os valores do campo de agregação forem `false`, MAXIMUM será 0.
  - Se todos os valores do campo de agregação forem `true`, MAXIMUM será 1.
  - Se os valores do campo de agregação forem uma mistura de `false` e `true`, MAXIMUM será 1.
- SUM é a soma do equivalente inteiro dos valores booleanos.
- COUNT é a contagem de coisas que correspondem aos critérios de string de consulta e contêm um valor de campo de agregação válido.

GetStatistics com valores numéricos

Quando você chama `GetStatistics` e especifica um campo de agregação do tipo `Number`, `GetStatistics` retorna os seguintes valores:

count

A contagem de itens que correspondem aos critérios de sequência de consulta e contêm um valor de campo de agregação válido.

média

A média dos valores numéricos que correspondem à consulta.

soma

A soma dos valores numéricos que correspondem à consulta.

mínimo

O menor dos valores numéricos que correspondem à consulta.

máximo

O maior dos valores numéricos que correspondem à consulta.

sumOfSquares

A soma dos quadrados dos valores numéricos que correspondem à consulta.

variância

A variação dos valores numéricos que correspondem à consulta. A variância de um conjunto de valores é a média dos quadrados das diferenças de cada valor em relação ao valor médio do conjunto.

stdDeviation

O desvio padrão dos valores numéricos que correspondem à consulta. O desvio padrão de um conjunto de valores é uma medida de como os valores estão distribuídos.

O exemplo a seguir mostra como chamar get-statistics com um campo personalizado numérico.

```
aws iot get-statistics --aggregation-field 'attributes.numericAttribute2'
            --query-string '*'
```

```
{
  "statistics": {
    "count": 3,
    "average": 33.333333333333336,
    "sum": 100.0,
    "minimum": -125.0,
    "maximum": 150.0,
    "sumOfSquares": 43750.0,
    "variance": 13472.22222222222,
    "stdDeviation": 116.06990230986766
  }
}
```

Para campos de agregação numéricos, se os valores de campo excederem o valor duplo máximo, os valores estatísticos estarão vazios

GetCardinality

A API GetCardinality e o comando get-cardinality da CLI retornam a contagem aproximada de valores exclusivos que correspondem à consulta. Por exemplo, você pode querer encontrar o número de dispositivos com níveis de bateria inferiores a 50%:

```
aws iot get-cardinality --index-name AWS_Things --query-string "batterylevel
          > 50" --aggregation-field "shadow.reported.batterylevel"
```

Este comando retorna o número de itens com níveis de bateria superiores a 50%:

```
{
    "cardinality": 100
}
```

`cardinality` é sempre retornado por get-cardinality mesmo se não houver campos correspondentes. Por exemplo:

```
aws iot get-cardinality --query-string "thingName:Non-existent*"
          --aggregation-field "attributes.customField_STR"
```

```
{
    "cardinality": 0
}
```

O comando da CLI get-cardinality usa os seguintes parâmetros:

`index-name`

O nome do índice a ser pesquisado. O valor padrão é `AWS_Things`.

`query-string`

A consulta usada para pesquisar o índice. É possível especificar`"*"`Para obter a contagem de todas as coisas indexadas em seuConta da AWS.

`aggregationField`

O campo a ser agregado.

`query-version`

A versão da consulta a ser usada. O valor padrão é `2017-09-30`.

GetPercentiles

A API GetPercentiles e o comando get-percentiles da CLI agrupam os valores agregados que correspondem à consulta em agrupamentos de percentil. Os agrupamentos de percentil padrão são: 1, 5, 25, 50, 75, 95, 99, embora você possa especificar o seu próprio quando chamar `GetPercentiles`. Esta função retorna um valor para cada grupo de percentis especificado (ou os agrupamentos de percentil padrão). O grupo de percentis “1" contém o valor de campo agregado que ocorre em aproximadamente um por cento dos valores que correspondem à consulta. O grupo de percentil “5" contém o valor de campo agregado que ocorre em aproximadamente cinco por cento dos valores que correspondem à consulta, e assim por diante. O resultado é uma aproximação. Quanto mais valores correspondem à consulta, mais precisos os valores do percentil.

O exemplo a seguir mostra como chamar o comando get-percentiles da CLI.

```
aws iot get-percentiles --query-string "thingName:*" --aggregation-field
          "attributes.customField_NUM" --percents 10 20 30 40 50 60 70 80 90 99
```

```
{
    "percentiles": [
        {
            "value": 3.0,
            "percent": 80.0
        },
        {
            "value": 2.5999999999999996,
            "percent": 70.0
        },
        {
            "value": 3.0,
            "percent": 90.0
        },
        {
```

```
            "value": 2.0,
            "percent": 50.0
        },
        {
            "value": 2.0,
            "percent": 60.0
        },
        {
            "value": 1.0,
            "percent": 10.0
        },
        {
            "value": 2.0,
            "percent": 40.0
        },
        {
            "value": 1.0,
            "percent": 20.0
        },
        {
            "value": 1.4,
            "percent": 30.0
        },
        {
            "value": 3.0,
            "percent": 99.0
        }
    ]
}
```

O comando a seguir mostra a saída retornada de get-percentiles quando não há documentos correspondentes.

```
aws iot get-percentiles --query-string "thingName:Non-existent*"
          --aggregation-field "attributes.customField_NUM"
```

```
{
    "percentiles": []
}
```

O comando da CLI get-percentile usa os seguintes parâmetros:

`index-name`

O nome do índice a ser pesquisado. O valor padrão é `AWS_Things`.

`query-string`

A consulta usada para pesquisar o índice. É possível especificar`"*"`Para obter a contagem de todas as coisas indexadas em seuConta da AWS.

`aggregationField`

O campo a ser agregado, que deve ser do tipo `Number`.

`query-version`

A versão da consulta a ser usada. O valor padrão é `2017-09-30`.

`percents`

Opcional. Você pode usar esse parâmetro para especificar agrupamentos de percentil personalizados.

GetBucketsAggregation

OGetBucketsAggregationAPI e oget-buckets-aggregationO comando da CLI retorna uma lista de buckets e o número total de coisas que se encaixam nos critérios de string de consulta.

O exemplo a seguir mostra como chamar o comando da CLI get-buckets-aggregation.

```
aws iot get-buckets-aggregation --query-string '*' --index-name AWS_Things --
aggregation-field 'shadow.reported.batterylevelpercent' --buckets-aggregation-type
 'termsAggregation={maxBuckets=5}'
```

Esse comando retorna o seguinte:

```
{
    "totalCount": 20,
    "buckets": [
        {
            "keyValue": "100",
            "count": 12
        },
        {
            "keyValue": "90",
            "count": 5
        },
        {
            "keyValue": "75",
            "count": 3
        }
    ]
}
```

O comando get-buckets-aggregation da CLI usa os seguintes parâmetros:

`index-name`

O nome do índice a ser pesquisado. O valor padrão é `AWS_Things`.

`query-string`

A consulta usada para pesquisar o índice. É possível especificar`"*"`Para obter a contagem de todas as coisas indexadas em seuConta da AWS.

`aggregation-field`

O campo a ser agregado.

`buckets-aggregation-type`

O controle básico da forma de resposta e do tipo de agregação de bucket a ser executado.

Autorização

Você pode especificar o índice de grupo de coisas como um ARN do recurso em uma ação de política da AWS IoT, da seguinte forma.

Ação	Recurso
`iot:GetStatistics`	O ARN de um índice (por exemplo, `arn:aws:iot:`*`your-aws-region`*`:index/`

Ação	Recurso
	`AWS_Things` ou `arn:aws:iot:`*`your-aws-region`*`:index/AWS_ThingGroups`).

Sintaxe de consulta

Na indexação de frota, você usa uma sintaxe de consulta para especificar consultas.

Recursos compatíveis

A sintaxe de consulta dá suporte aos seguintes recursos:

- Termos e frases
- Pesquisar campos
- Pesquisa de prefixo
- Pesquisa de intervalo
- Operadores boolianos `AND`, `OR`, `NOT` e `-`. O hífen é usado para excluir algo dos resultados de pesquisa (por exemplo,`thingName:(tv* AND -plasma)`).
- Agrupamento
- Agrupamento de campos
- Escape de caracteres especiais (como\)

Recursos do não compatíveis

A sintaxe da consulta não é compatível com os seguintes recursos do:

- Pesquisa com caracteres curinga iniciais (como "*xyz"), mas a pesquisa de "*" corresponde a todas as coisas
- Expressões regulares
- Aumento
- Classificação
- Pesquisas difusas
- Pesquisa de proximidade
- Classificar
- Agregação

Observações

Alguns comentários sobre a linguagem de consulta:

- O operador padrão é AND. Uma consulta de `"thingName:abc thingType:xyz"` é equivalente a `"thingName:abc AND thingType:xyz"`.
- Se um campo não for especificado,AWS IoTPesquisa o termo em todos os campos do Registro, do Device Shadow e do Device Defender.
- Todos os nomes de campos diferenciam maiúsculas de minúsculas.
- A pesquisa não diferencia maiúsculas de minúsculas. As palavras são separadas por caracteres de espaço em branco, conforme definido pelo Java`Character.isWhitespace(int)`.

- A indexação de dados do Device Shadow (sombras sem nome e sombras nomeadas) inclui seções relatadas, desejadas, delta e de metadados.
- As versões do device shadow e do registry não são pesquisáveis, mas estão presentes na resposta.
- O número máximo de termos em uma consulta é sete.

Exemplo de consultas de coisas

> **Note**
>
> O recurso de indexação de frota para suportar indexação denominada sombras eAWS IoT Device DefenderOs dados de violações estão em versão de visualização paraAWS IoT Device Managemente sujeita a alterações.

Especifique consultas em uma string de consulta usando a sintaxe de consulta. As consultas são passadas para o`SearchIndex`API. A tabela a seguir lista alguns exemplos de sequências de consulta.

String de consulta	Result
abc	Consultas para “abc” em qualquer registro, sombra (sombra clássica sem nome e sombra nomeada) ou campo de violações do Device Defender.
thingName:myThingName	Consulta uma coisa com o nome "myThingName".
thingName:my*	Consulta coisas com nomes que começam com "my".
thingName:ab?	Consulta coisas com nomes que têm “ab” e um caractere adicional (por exemplo, “aba”, “abb”, “abc” e assim por diante.)
thingTypeName:aa	Consultas sobre coisas que estão associadas ao tipo “aa”.
attributes.myAttribute:75	Consulta coisas com um atributo chamado "myAttribute" que tem o valor 75.
attributes.myAttribute:[75 TO 80]	Consulta coisas com um atributo chamado “myAttribute” que tem um valor que está dentro de um intervalo numérico (75—80, inclusive).
attributes.myAttribute:{75 TO 80]	Consulta coisas com um atributo chamado “myAttribute” que tem um valor que está dentro do intervalo numérico (>75 e <=80).
attributes.serialNumber:["abcd" TO "abcf"]	Consulta coisas com um atributo chamado “serialNumber” que tem um valor dentro de um intervalo de sequência alfanumérica. Essa consulta retornará coisas com um atributo "serialNumber" com valores "abcd", "abce" ou "abcf".
attributes.myAttribute:i*t	Consulta coisas com um atributo chamado “myAttribute” onde o valor é 'i', seguido por qualquer número de caracteres, seguido por 't'.

String de consulta	Result
attributes.attr1:abc AND attributes.attr2<5 NOT attributes.attr3>10	Consulta coisas que combinam termos usando expressões boolianas. Essa consulta retorna coisas que têm um atributo chamado "attr1" com um valor "abc", um atributo chamado "attr2" que é menor que 5 e um atributo chamado "attr3" que não é maior que 10.
shadow.hasDelta:true	Consulta coisas com uma sombra sem nome que tem um elemento delta.
NOT attributes.model:legacy	Consulta de coisas em que o atributo denominado "modelo" não é "legado".
shadow.reported.stats.battery:{70 TO 100} (v2 OR v3) NOT attributes.model:legacy	Consulta coisas com o seguinte: • O atributo `stats.battery` de sombra da coisa tem um valor entre 70 e 100. • O texto "v2" ou "v3" ocorre em um nome de coisa, nome de tipo ou valores do atributo. • O atributo `model` da coisa não é definido como "legado".
shadow.reported.myvalues:2	Consulta coisas onde a matriz `myvalues` na seção relatada da shadow contém um valor igual a 2.
shadow.reported.location:* NOT shadow.desired.stats.battery:*	Consulta coisas com o seguinte: • O atributo `location` existe na seção `reported` da shadow. • O`stats.battery`O atributo não existe no`desired`seção.
shadow.name. <shadowName>.HASDELTA: True	Consultas para coisas que têm uma sombra com o nome dado e também um elemento delta. O recurso de indexação de frota para suportar indexação denominada shadow está na versão de visualização paraAWS IoTGerenciamento de dispositivos.
shadow.name. <shadowName>.desired.filamento: *	Consultas para coisas que têm uma sombra com o nome dado e também uma propriedade de filamento desejada. O recurso de indexação de frota para suportar indexação de sombras nomeadas está na versão prévia paraAWS IoTGerenciamento de dispositivos.
shadow.name. <shadowName>.reported.location: *	Consultas para coisas que têm uma sombra com o nome dado e onde `o`locationO atributo existe na seção relatada da shadow nomeada. O recurso de indexação de frota para suportar indexação de sombras nomeadas está na versão prévia paraAWS IoTGerenciamento de dispositivos.
connectivity.connected:true	Consultas para todos os dispositivos conectados.

String de consulta	Result
connectivity.connected:false	Consultas para todos os dispositivos desconectados.
connectivity.connected:true AND connectivity.timestamp: [1557651600000 PARA 1557867600000]	Consultas para todos os dispositivos conectados com um registro de data/hora de conexão >= 1557651600000 and <= 1557867600000. Os registro de data e hora são expressos em milissegundos desde o epoch.
connectivity.connected:false AND connectivity.timestamp: [1557651600000 PARA 1557867600000]	Consultas para todos os dispositivos desconectados com um registro de data/hora de desconexão >= 1557651600000 and <= 1557867600000. Os registro de data e hora são expressos em milissegundos desde o epoch.
connectivity.connected:true AND connectivity.timestamp > 1557651600000	Consultas para todos os dispositivos conectados com um registro de data/hora de conexão > 1557651600000. Os registro de data e hora são expressos em milissegundos desde o epoch.
connectivity.connected:*	Consultas para todos os dispositivos com informações de conectividade presentes.
Connectivity.DisconnectReason: *	Consulta todos os dispositivos com conectividade DisconnectReason presente.
Connectivity.disconnectReason:CLIENT_INITIATED_DISCONNECT	Consultas para todos os dispositivos desconectados devido a CLIENT_INITIATED_DISCONNECT.
DeviceDefender.violationCount: [0 A 100]	Consulta coisas com um valor de violação do Device Defender contam o valor que está dentro do intervalo numérico (0-100, inclusive). O recurso de indexação de frota para suportar a indexação de dados de violações do Device Defender está em versão prévia paraAWS IoTGerenciamento de dispositivos.
DeviceDefender. <device-SecurityProfile>.DisconnectBehavior.Inviolation:True	Consultas para coisas que são violados pelo comportamento`disconnectBehavior`conforme definido no perfil de segurança`device-SecurityProfile`. Observe queInviolação:falsonão é uma consulta válida. O recurso de indexação de frota para suportar a indexação de dados de violações do Device Defender está em versão prévia paraAWS IoTGerenciamento de dispositivos.

String de consulta	Result
DeviceDefender. <device-SecurityProfile>.DisconnectBehavior.LastViolationValue.Number=2	Consultas para coisas que são violados pelo comportamento `disconnectBehavior` conforme definido no perfil de segurança Device-SecurityProfile com um último valor de evento de violação maior que 2. O recurso de indexação de frota para suportar a indexação de dados de violações do Device Defender está em versão prévia para AWS IoT Gerenciamento de dispositivos.
DeviceDefender. <device-SecurityProfile>.DisconnectBehavior.LastViolationTime>=1634297000000	Consultas para coisas que são violados pelo comportamento `disconnectBehavior` conforme definido no perfil de segurança Device-SecurityProfile com um último evento de violação após uma época especificada. O recurso de indexação de frota para suportar a indexação de dados de violações do Device Defender está em versão prévia para AWS IoT Gerenciamento de dispositivos.

Exemplo de consultas de grupos de coisas

As consultas são especificadas em uma sequência de consulta usando uma sintaxe de consulta e passadas para a API `SearchIndex`. A tabela a seguir lista alguns exemplos de sequências de consulta.

String de consulta	Result
abc	Consultas para "abc" em qualquer campo.
ThingGroupNames:MyGroupThingName	Consulta um grupo de coisas com o nome "myGroupThingName".
ThingGroupnames:Minha*	Consulta grupos de coisas com nomes que começam com "my".
CoisasNomes do grupo: AB?	Consulta grupos de coisas com nomes que têm "ab" e um caractere adicional (por exemplo: "aba", "abb", "abc" etc.)
attributes.myAttribute:75	Consulta grupos de coisas com um atributo chamado "myAttribute" que tem o valor 75.
attributes.myAttribute:[75 TO 80]	Consulta grupos de coisas com um atributo chamado “myAttribute” cujo valor está dentro de um intervalo numérico (75—80, inclusive).
attributes.myAttribute:[75 TO 80]	Consulta grupos de coisas com um atributo chamado "myAttribute" cujo valor está dentro do intervalo numérico (>75 e <=80).
attributes.myAttribute:["abcd" TO "abcf"]	Consulta grupos de coisas com um atributo chamado "myAttribute" cujo valor está dentro de um intervalo de sequência alfanumérica. Essa consulta retornará grupos de coisas com um

String de consulta	Result
	atributo "serialNumber" com valores "abcd", "abce" ou "abcf".
attributes.myAttribute:i*t	Consulta grupos de coisas com um atributo chamado "myAttribute" cujo valor é 'i', seguido por qualquer número de caracteres, seguido por 't'.
attributes.attr1:abc AND attributes.attr2<5 NOT attributes.attr3>10	Consulta grupos de coisas que combinam termos usando expressões boolianas. Essa consulta retorna grupos de coisas que têm um atributo chamado "attr1" com um valor "abc", um atributo chamado "attr2" que é menor que 5 e um atributo chamado "attr3" que não é maior que 10.
NOT attributes.myAttribute:cde	Consulta grupos de coisas onde o atributo chamado "myAttribute" não é "cde".
parentGroupNames:(myParentThingGroupName)	Consulta grupos de coisas cujo nome do grupo pai corresponde a "myParentThingGroupName".
parentGroupNames:(myParentThingGroupName OR myRootThingGroupName)	Consulta grupos de coisas cujo nome do grupo pai corresponde a "myParentThingGroupName" ou "myRootThingGroupName".
parentGroupNames:(myParentThingGroupNa*)	Consulta grupos de coisas cujo nome do grupo pai se inicia com "myParentThingGroupNa".

Métricas de frota

Métricas de frota são um recurso doindexação de frota (p. 836), um serviço gerenciado que permite indexar, pesquisar e agregar os dados de seus dispositivos emAWS IoT. Com as métricas de frota, você pode monitorar o estado agregado de seus dispositivos de frota emCloudWatchao longo do tempo, incluindo a revisão da taxa de desconexão dos dispositivos da frota ou as alterações médias no nível da bateria de um período especificado.

Usando métricas de frota, você pode criarConsultas de agregação (p. 851)cujos resultados são emitidos continuamente paraCloudWatchcomo métricas para analisar tendências e criar alarmes. Para suas tarefas de monitoramento, você pode especificar as consultas de agregação de diferentes tipos de agregação (Estatísticas,Cardinality, ePercentil). Você pode salvar todas as consultas de agregação para criar métricas de frota para reutilização no futuro.

Tutorial de conceitos básicos

Neste tutorial, você cria umMétrica de fro (p. 862)para monitorar as temperaturas dos seus sensores para detectar possíveis anomalias. Ao criar a métrica da frota, você define umConsulta de agregação (p. 851)que detecta o número de sensores com temperaturas superiores a 80 graus Fahrenheit. Você especifica a consulta a ser executada a cada 60 segundos e os resultados da consulta são emitidos para o CloudWatch, onde você pode visualizar o número de sensores que têm riscos potenciais de alta temperatura e definir alarmes. Para concluir este tutorial, você usaráAWS CLI.

Neste tutorial, você saberá como:

- Configurar (p. 863)
- Criar métricas de frota (p. 864)
- Exibir métricas no CloudWatch (p. 866)

- Limpar os recursos (p. 867)

Esse tutorial leva cerca de 15 minutos para ser concluído.

Pré-requisitos

- Instalar a versão mais recente do AWS CLI
- Familiarize-se com oConsulta de dados agregados
- Familiarize-se com oComo usar métricas do Amazon CloudWatch

Configurar

Para usar métricas de frota, você deve ativar a indexação da frota. Para habilitar a indexação de frota para suas coisas ou grupos de coisas com fontes de dados especificadas e configurações associadas, siga as instruções emGerenciar a indexação de coisas (p. 840)eGerenciar a indexação de grupos de coisas (p. 849).

Para configurar a

1. Execute o comando a seguir para habilitar a indexação da frota e especificar as fontes de dados para pesquisar.

```
aws iot update-indexing-configuration \
--thing-indexing-configuration
 "thingIndexingMode=REGISTRY_AND_SHADOW,customFields=[{name=attributes.temperature,type=Number},
{name=attributes.rackId,type=String},
{name=attributes.stateNormal,type=Boolean}],thingConnectivityIndexingMode=STATUS" \
```

   O comando de CLI de exemplo acima permite a indexação de frota para oferecer suporte à pesquisa de dados de registro, de dados de sombra e o status de conectividade da coisa usando o`AWS_Things`INDEX.

   A alteração de configuração pode levar alguns minutos para ser concluída. Verifique se a indexação da frota está ativada antes de criar métricas de frota.

   Para verificar se a indexação de frota foi ativada, execute o seguinte comando da CLI:

```
aws --region us-east-1 iot describe-index --index-name "AWS_Things"
```

   Para obter mais informações, leiaHabilitar indexação de coisas (p. 840).

2. Execute o seguinte script bash para criar dez coisas e descrevê-las.

```
# Bash script. Type `bash` before running in other shells.

Temperatures=(70 71 72 73 74 75 47 97 98 99)
Racks=(Rack1 Rack1 Rack2 Rack2 Rack3 Rack4 Rack5 Rack6 Rack6 Rack6)
IsNormal=(true true true true true true false false false false)

for ((i=0; i < 10; i++))
do
  thing=$(aws iot create-thing --thing-name "TempSensor$i" --attribute-payload
 attributes="{temperature=${Temperatures[@]:$i:1},rackId=${Racks[@]:$i:1},stateNormal=
${IsNormal[@]:$i:1}}")
  aws iot describe-thing --thing-name "TempSensor$i"
done
```

Esse script cria dez coisas para representar dez sensores. Cada coisa tem atributos de`temperature`,`rackId`, `estateNormal`conforme descrito na tabela a seguir:

Atributo	Tipo de dados	Descrição
`temperature`	telefone	Valor da temperatura em Fahrenheit
`rackId`	String	ID do rack do servidor que contém sensores
`stateNormal`	Booliano	Clima o valor da temperatura do sensor é normal ou não

A saída desse script contém dez arquivos JSON. Um dos arquivos JSON é semelhante ao seguinte:

```
{
    "version": 1,
    "thingName": "TempSensor0",
    "defaultClientId": "TempSensor0",
    "attributes": {
        "rackId": "Rack1",
        "stateNormal": "true",
        "temperature": "70"
    },
    "thingArn": "arn:aws:iot:region:account:thing/TempSensor0",
    "thingId": "example-thing-id"
}
```

Para obter mais informações, leiaCriar uma coisa.

Criar métricas de frota

Para criar uma métrica de frota

1. Execute o comando a seguir para criar uma métrica de frota chamada*`High_temp_FM`*. Você cria a métrica da frota para monitorar o número de sensores com temperaturas superiores a 80 graus Fahrenheit no CloudWatch.

```
aws iot create-fleet-metric --metric-name "high_temp_FM" --query-string
 "thingName:TempSensor* AND attributes.temperature >80" --period 60 --aggregation-field
 "attributes.temperature" --aggregation-type name=Statistics,values=count
```

—métrico-name

Tipo de dados: string. O`--metric-name`parâmetro especifica um nome de métrica de frota. Neste exemplo, você está criando uma métrica de frota chamadaHigh_temp_FM.

—query-string

Tipo de dados: string. O`--query-string`parâmetro especifica a string de consulta. Neste exemplo, a string de consulta significa consultar todas as coisas com nomes começando comTempSensore com temperaturas superiores a 80 graus Fahrenheit. Para obter mais informações, leiaSintaxe de consulta (p. 857).

—period

Tipo de dados: inteiro. O`--period`parâmetro especifica o tempo para recuperar os dados agregados em segundos. Neste exemplo, você especifica que a métrica da frota que você está criando recupera os dados agregados a cada 60 segundos.

—campo de agregação

Tipo de dados: string. O`--aggregation-field`parâmetro especifica o atributo a ser avaliado. Neste exemplo, o atributo temperature deve ser avaliado.

—tipo de agregação

O`--aggregation-type`parâmetro especifica o resumo estatístico a ser exibido na métrica da frota. Para suas tarefas de monitoramento, você pode personalizar as propriedades de consulta de agregação para os diferentes tipos de agregação (Estatísticas,Cardinality, ePercentil). Neste exemplo, você especificacontagempara o tipo de agregaçãoEstatísticasPara retornar a contagem de dispositivos que têm atributos que correspondem à consulta, em outras palavras, para retornar a contagem dos dispositivos com nomes começando comTempSensore com temperaturas superiores a 80 graus Fahrenheit. Para obter mais informações, leiaConsulta de dados agregados (p. 851).

A saída desse comando é semelhante à seguinte:

```
{
    "metricArn": "arn:aws:iot:region:111122223333:fleetmetric/high_temp_FM",
    "metricName": "high_temp_FM"
}
```

> **Note**
>
> Pode levar um momento para que os pontos de dados sejam exibidos no CloudWatch.

Para saber mais sobre como criar uma métrica de frota, leiaGerenciamento de métricas de frota (p. 867).

Se você não conseguir criar uma métrica de frota, leiaSolução de problemas de métricas de frota (p. 1289).

2. (Opcional) Execute o comando a seguir para descrever a métrica de frota chamadaHigh_temp_FM:

```
aws iot describe-fleet-metric --metric-name "high_temp_FM"
```

A saída desse comando é semelhante à seguinte:

```
{
    "queryVersion": "2017-09-30",
    "lastModifiedDate": 1625249775.834,
    "queryString": "*",
    "period": 60,
    "metricArn": "arn:aws:iot:region:111122223333:fleetmetric/high_temp_FM",
    "aggregationField": "registry.version",
    "version": 1,
    "aggregationType": {
        "values": [
            "sum"
        ],
        "name": "Statistics"
    },
    "indexName": "AWS_Things",
```

```
    "creationDate": 1625249775.834,
    "metricName": "high_temp_FM"
}
```

Exibir métricas de frota no CloudWatch

Depois de criar a métrica de frota, você pode exibir os dados de métrica no CloudWatch. Neste tutorial, você verá a métrica que mostra o número de sensores com nomes começando comTempSensore com temperaturas superiores a 80 graus Fahrenheit.

Para visualizar pontos de dados no CloudWatch

1. Abra o console do CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No menu do CloudWatch à esquerda, escolha o escolhaMétricasPara expandir o submenu e escolhaTodas as métricas. Isso abre a página com a metade superior para exibir o gráfico e a metade inferior contendo quatro seções com guias.
3. A primeira seção com guiasTodas as métricaslista todas as métricas que você pode visualizar em grupos, escolhaIOTFleetMetricsque contém todas as métricas da sua frota.
4. NoTipo de agregaçãoSeção do doTodas as métricasGuia, selecioneTipo de agregaçãopara visualizar todas as métricas de frota que você criou.
5. Escolha a métrica da frota para exibir o gráfico à esquerda doTipo de agregaçãoseção. Você verá o valor*soma*à esquerda doNome da métrica, e esse é o valor do tipo de agregação que você especificou noCriar métricas de frota (p. 864)Seção deste tutorial.
6. Selecione a segunda guia chamadaMétricas em gráficosà direita doTodas as métricasPara exibir a métrica de frota que você escolheu da etapa anterior.

   Você deve ser capaz de ver um gráfico que exibe o número de sensores com temperaturas superiores a 80 graus Fahrenheit como o seguinte:

Note

OPeríodoO atributo no CloudWatch é padrão de 5 minutos. É o intervalo de tempo entre os pontos de dados exibidos no CloudWatch. Você pode alterar oPeríodoDefinir com base em suas necessidades.

7. (Opcional) Você pode definir um alarme métrico.
   1. No menu do CloudWatch à esquerda, escolha o escolhaAlarmesPara expandir o submenu e escolhaTodos os alarmes.
   2. NoAlarmesPágina, selecioneCriar alarmeno canto superior direito. Siga oCriar alarmeinstruções no console para criar um alarme conforme necessário. Para obter mais informações, leiaUsar alarmes do Amazon CloudWatch.

Para saber mais, leiaComo usar métricas do Amazon CloudWatch.

Se você não conseguir ver pontos de dados no CloudWatch, leiaSolução de problemas de métricas de frota (p. 1289).

Limpar

Para excluir métricas de frota

Você usa odelete-fleet-metricComando da CLI para excluir métricas de frota.

Execute o comando a seguir para excluir a métrica de frota chamadaHigh_temp_FM.

```
aws iot delete-fleet-metric --metric-name "high_temp_FM"
```

Para limpar as coisas

Você usa odelete-thingComando da CLI para excluir coisas.

Execute o script a seguir para excluir as dez coisas que você criou:

```
# Bash script. Type `bash` before running in other shells.

for ((i=0; i < 10; i++))
do
  thing=$(aws iot delete-thing --thing-name "TempSensor$i")
done
```

Para limpar métricas no CloudWatch

O CloudWatch não é compatível com exclusão de métricas. As métricas expiram com base em seus cronogramas de retenção. Para saber mais, leiaComo usar métricas do Amazon CloudWatch.

Gerenciamento de métricas de frota

Este tópico mostra como usar oAWS IoTConsole eAWS CLIpara gerenciar as métricas de sua frota.

Gerenciamento de métricas de frota (console)

Certifique-se de ter habilitado a indexação de frota com fontes de dados e configurações associadas antes de criar métricas de frota.

Habilitar a indexação de

Se você já ativou a indexação da frota, pule esta seção.

Se você não habilitou a indexação de frota, siga estas instruções.

1. Abra oAWS IoTConsole nohttps://console.aws.amazon.com/iot/.
2. NoAWS IoTMenu, selecioneConfigurações.
3. Para visualizar as configurações detalhadas, noConfiguraçõespágina, role até oIndexação de frotaseção.
4. Para atualizar as configurações de indexação da frota, à direita doIndexação de frotaseção, selecioneGerenciar a indexação.
5. NoGerenciamento de indexação de frota, atualize as configurações de indexação da frota com base em suas necessidades.
   - Configuração

     Para ativar a indexação de coisas, alterneIndexação de coisason e, em seguida, selecione as fontes de dados das quais você deseja indexar.

     Para ativar a indexação de grupos de coisas, alterneIndexação de grupo de coisasligado.
   - Campos de pesquisa personalizados -opcionais

     Campos de pesquisa personalizados são uma lista de pares de nome de campo e tipo de campo.

     Para adicionar um par de campos personalizado, escolhaAdicionar novo campo. Digite um nome de campo personalizado, como`attributes.temperature`e, em seguida, selecione um tipo de campo naTipo de campomenu. Observe que um nome de campo personalizado começa com:`attributes.`e será salvo como um atributo a ser executadoConsultas de agregações de coisas.

     Para atualizar e salvar a configuração, escolhaAtualização.

Criar uma métrica de frota

1. Abra oAWS IoTConsole nohttps://console.aws.amazon.com/iot/.
2. NoAWS IoTMenu, selecioneGerenciee depois escolhaMétricas de frota.
3. NoMétricas de frotaPágina, selecioneCriar métrica de frotae conclua as etapas de criação.
4. Na etapa 1Configurar métricas de frota
   - DentroConsulte, insira uma string de consulta para especificar as coisas ou grupos de coisas que você deseja realizar a pesquisa agregada. A string de consulta consiste em um atributo e um valor. para oProperties, escolha o atributo desejado ou, se ele não aparecer na lista, insira o atributo no campo. Insira o valor após`:`. Uma string de consulta de exemplo pode ser`thingName:TempSensor*`. Para cada string de consulta inserida, pressioneentrarNo teclado. Se você inserir várias strings de consulta, especifique seu relacionamento selecionandoe,ou,e não, ouou nãoentre eles.
   - DentroPropriedades do relatório, escolhaNome do índice,Tipo de agregação, eAgregação de campode suas respectivas listas. Em seguida, selecione os dados nos quais você deseja agregarSelecionar dados, onde você pode selecionar vários valores de dados.
   - Escolha Next (Próximo).
5. Na etapa 2Especificar as propriedades da métrica
   - DentroNome da métrica do, insira um nome para a métrica de frota que você está criando.
   - DentroDescrição -opcionais, insira uma descrição para a métrica de frota que você está criando. Esse campo é opcional.

- DentroHoraseminutos, insira a hora (com que frequência) você deseja que a métrica da frota emita dados para o CloudWatch.
- Escolha Next (Próximo).

6. Na etapa 3Revisar e criar
   - Revise as configurações da etapa 1 e da etapa 2. Para editar as configurações, escolhaEdite.
   - SelecioneCriar métrica de frota.

Após a criação bem-sucedida, a métrica da frota é listada noMétrica do.

Atualizar uma métrica de fro

1. NoMétrica doNa página, escolha a métrica de frota que deseja atualizar.
2. Na métrica de frotaDetalhesPágina, selecioneEdite. Isso abre as etapas de criação nas quais você pode atualizar a métrica da frota em qualquer uma das três etapas.
3. Depois de concluir a atualização da métrica da frota, escolhaAtualizar métrica de fro.

Exclua uma métrica do

1. NoMétrica doNa página, escolha a métrica de frota que deseja excluir.
2. Na próxima página que mostra detalhes da métrica da sua frota, escolhaExcluir.
3. Na caixa de diálogo, insira o nome da métrica da frota para confirmar a exclusão.
4. Escolha Delete (Excluir). Esta etapa exclui a métrica da frota permanentemente.

Gerenciamento de métricas de frota (CLI)

As seções a seguir mostram como usar oAWSCLI para gerenciar as métricas da sua frota. Certifique-se de ter habilitado a indexação de frota com fontes de dados e configurações associadas antes de criar métricas de frota. Para ativar a indexação de frota para suas coisas ou grupos de coisas, siga as instruções emGerenciar a indexação de coisas (p. 840)ouGerenciar a indexação de grupos de coisas (p. 849).

Criar uma métrica de frota

Você usa o comando create-fleet-metric CLI para criar uma métrica de frota.

```
aws iot create-fleet-metric --metric-name "YourFleetMetricName" --query-string "*" --period
 60 --aggregation-field "registry.version" --aggregation-type name=Statistics,values=sum
```

A saída desse comando contém o nome e o nome de recurso da Amazon (ARN) da métrica de sua frota. A saída será exibida como a seguir:

```
{
    "metricArn": "arn:aws:iot:us-east-1:111122223333:fleetmetric/YourFleetMetricName",
    "metricName": "YourFleetMetricName"
}
```

Listar métricas de frota

Você usa o comando list-fleet-metric CLI para listar todas as métricas da frota em sua conta.

```
aws iot list-fleet-metrics
```

A saída desse comando contém todas as métricas de frota. A saída será exibida como a seguir:

```
{
    "fleetMetrics": [
        {
            "metricArn": "arn:aws:iot:us-east-1:111122223333:fleetmetric/
YourFleetMetric1",
            "metricName": "YourFleetMetric1"
        },
        {
            "metricArn": "arn:aws:iot:us-east-1:111122223333:fleetmetric/
YourFleetMetric2",
            "metricName": "YourFleetMetric2"
        }
    ]
}
```

Descreva uma métrica de

Use o comando describe-fleet-metric CLI para exibir informações mais detalhadas sobre uma métrica de frota.

```
aws iot describe-fleet-metric --metric-name "YourFleetMetricName"
```

A saída do comando contém as informações detalhadas sobre a métrica de frota especificada. A saída será exibida como a seguir:

```
{
    "queryVersion": "2017-09-30",
    "lastModifiedDate": 1625790642.355,
    "queryString": "*",
    "period": 60,
    "metricArn": "arn:aws:iot:us-east-1:111122223333:fleetmetric/YourFleetMetricName",
    "aggregationField": "registry.version",
    "version": 1,
    "aggregationType": {
        "values": [
            "sum"
        ],
        "name": "Statistics"
    },
    "indexName": "AWS_Things",
    "creationDate": 1625790642.355,
    "metricName": "YourFleetMetricName"
}
```

Atualizar uma métrica de fro

Use o comando update-fleet-metric CLI para atualizar uma métrica de frota.

```
aws iot update-fleet-metric --metric-name "YourFleetMetricName" --query-string
 "*" --period 120 --aggregation-field "registry.version" --aggregation-type
 name=Statistics,values=sum,count --index-name AWS_Things
```

O comando update-fleet-metric não produz nenhuma saída. Você pode usar o comando describe-fleet-metric CLI para exibir o resultado.

```
{
    "queryVersion": "2017-09-30",
```

```
    "lastModifiedDate": 1625792300.881,
    "queryString": "*",
    "period": 120,
    "metricArn": "arn:aws:iot:us-east-1:111122223333:fleetmetric/YourFleetMetricName",
    "aggregationField": "registry.version",
    "version": 2,
    "aggregationType": {
        "values": [
            "sum",
            "count"
        ],
        "name": "Statistics"
    },
    "indexName": "AWS_Things",
    "creationDate": 1625792300.881,
    "metricName": "YourFleetMetricName"
}
```

Exclua uma métrica do

Use o comando delete-fleet-metric CLI para excluir uma métrica de frota.

```
aws iot delete-fleet-metric --metric-name "YourFleetMetricName"
```

Esse comando não produzirá nenhuma saída se a exclusão for bem-sucedida ou se você especificar uma métrica de frota que não existe.

Para obter mais informações, leiaSolução de problemas de métricas de frota (p. 1289).

Entrega de arquivos baseada em MQTT

Uma opção que você pode usar para gerenciar arquivos e transferi-los paraAWS IoTdispositivos em sua frota é a entrega de arquivos baseada em MQTT. Com este recurso noAWSNuvem você pode criar umfluxoque contém vários arquivos, você pode atualizar dados de fluxo (a lista de arquivos e descrições), obter os dados do fluxo e muito mais.AWS IoT A entrega de arquivos baseada em MQTT pode transferir dados em pequenos blocos para seus dispositivos IoT, usando o protocolo MQTT com suporte para mensagens de solicitação e resposta em JSON ou CBOR.

Para obter mais informações sobre formas de transferir dados de e para dispositivos IoT usandoAWS IoT, consulteConectar dispositivos aoAWS IoT (p. 79).

Tópicos

- O que é um fluxo? (p. 872)
- Gerenciando um fluxo noAWSNuvem (p. 873)
- O uso doAWS IoTEntrega de arquivos baseada em MQTT em dispositivos (p. 874)
- Um exemplo de caso de uso no FreeRTOS OTA (p. 881)

O que é um fluxo?

DentroAWS IoT, umfluxoé um recurso endereçável publicamente que é uma abstração para uma lista de arquivos que podem ser transferidos para um dispositivo IoT. Um fluxo típico contém as seguintes informações:

- Um Nome do Recurso da Amazon (ARN)que identifica exclusivamente um fluxo em um determinado momento. Este ARN tem o padrão`arn:`*`partition`*`:iot:`*`region`*`:`*`account-ID`*`:stream/`*`stream ID`*.
- Um ID do fluxoque identifica seu fluxo e é usado (e geralmente necessário) emAWS Command Line Interface(AWS CLI) ou comandos SDK.
- Uma descrição de fluxoIsso fornece uma descrição do recurso de fluxo.
- Uma versão em streamque identifica uma versão específica do fluxo. Como os dados de fluxo podem ser modificados imediatamente antes que os dispositivos iniciem a transferência de dados, a versão do stream pode ser usada pelos dispositivos para impor uma verificação de consistência.
- Uma lista de arquivosque podem ser transferidos para dispositivos. Para cada arquivo na lista, o stream registra um ID de arquivo, o tamanho do arquivo e as informações de endereço do arquivo, que consistem, por exemplo, no nome do bucket do Amazon S3, na chave do objeto e na versão do objeto.
- UmaAWS Identity and Access ManagementFunção (IAM)que concedemAWS IoTO arquivo baseado em MQTT fornece a permissão para ler arquivos de fluxo armazenados no armazenamento de dados.

AWS IoTA entrega de arquivos baseada em MQTT fornece a seguinte funcionalidade para que os dispositivos possam transferir dados doAWSNuvem:

- Transferência de dados usando o protocolo MQTT.
- Support para formatos JSON ou CBOR.

- A capacidade de descrever um fluxo (`DescribeStream`API) para obter uma lista de arquivos de stream, versão de fluxo e informações relacionadas.
- A capacidade de enviar dados em pequenos blocos (`GetStream`API) para que dispositivos com restrições de hardware possam receber os blocos.
- Support para um tamanho de bloco dinâmico por solicitação, para suportar dispositivos com capacidades de memória diferentes.
- Otimização para solicitações de streaming simultâneas quando vários dispositivos solicitam blocos de dados do mesmo arquivo de fluxo.
- Amazon S3 como armazenamento de dados para arquivos de stream.
- Support para publicação de log de transferência de dados deAWS IoTEntrega de arquivos com base em MQTT para o CloudWatch.

Para cotas de entrega de arquivos baseadas em MQTT, consulteAWS IoT CoreCotas de serviço donoAWSReferência geral.

Gerenciando um fluxo noAWSNuvem

AWS IoTforneceAWSSDK eAWS CLIcomandos que você pode usar para gerenciar um fluxo noAWSNuvem. Você pode usar esses comandos para fazer o seguinte:

- Crie um fluxo. CLI/SDK
- Descreva um fluxo para obter suas informações. CLI/SDK
- Listar fluxos noConta da AWS. CLI/SDK
- Atualize a lista de arquivos ou a descrição do fluxo em um fluxo. CLI/SDK
- Exclua um fluxo. CLI/SDK

> **Note**
>
> Neste momento, os fluxos não são visíveis noAWS Management Console. Você deve usar oAWS CLIouAWSSDK para gerenciar um fluxo noAWS IoT.

Antes de usarAWS IoTEntrega de arquivos baseada em MQTT de seus dispositivos, você deve seguir as etapas nas próximas seções para garantir que seus dispositivos estejam devidamente autorizados e possam se conectar aoAWS IoTGateway do dispositivo.

Conceder permissões para seus dispositivos

Você pode seguir as etapas emCriar umAWS IoTregrapara criar uma política de dispositivo ou usar uma política de dispositivo existente. Anexe a política aos certificados associados aos dispositivos e adicione as seguintes permissões à política de dispositivo.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {   "Effect": "Allow",
            "Action": [ "iot:Connect" ],
            "Resource": [
                "arn:partition:iot:region:accountID:client/
${iot:Connection.Thing.ThingName}"
            ]
        },
        {
            "Effect": "Allow",
```

```
            "Action": [ "iot:Receive", "iot:Publish" ],
            "Resource": [
                "arn:partition:iot:region:accountID:topic/$aws/things/
${iot:Connection.Thing.ThingName}/streams/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": "iot:Subscribe",
            "Resource": [
                "arn:partition:iot:region:accountID:topicfilter/$aws/things/
${iot:Connection.Thing.ThingName}/streams/*"
            ]
        }
    ]
}
```

Connect os dispositivos aoAWS IoT

Dispositivos que usamAWS IoTA entrega de arquivos baseada em MQTT é necessária para se conectar comAWS IoT.AWS IoT A entrega de arquivos baseada em MQTT se integra comAWS IoTnoAWSNuvem, portanto, seus dispositivos devem se conectar diretamente aoo endpoint doAWS IoTPlano de dados do.

> **Note**
>
> O endpoint doAWS IoTplano de dados é específico para oConta da AWSe Região. Você deve usar o endpoint para oConta da AWSe a região na qual seus dispositivos estão registrados emAWS IoT.

Consulte Como conectar-se ao AWS IoT Core (p. 71) para obter mais informações.

O uso doAWS IoTEntrega de arquivos baseada em MQTT em dispositivos

Para iniciar o processo de transferência de dados, um dispositivo deve receber umConjunto de dados iniciais, que inclui um ID de fluxo no mínimo. Você pode usar umTrabalhos (p. 681)para agendar tarefas de transferência de dados para seus dispositivos, incluindo o conjunto de dados inicial no documento de trabalho. Quando um dispositivo recebe o conjunto de dados inicial, ele deve iniciar a interação comAWS IoTEntrega de arquivos baseada em MQTT. Para trocar dados comAWS IoTEntrega de arquivos baseada em MQTT, um dispositivo deve:

- Use o protocolo MQTT para assinar oTópicos de entrega de arquivos baseados em MQTT (p. 116).
- Envie solicitações e aguarde para receber as respostas usando mensagens MQTT.

Opcionalmente, você pode incluir um ID de arquivo de fluxo e uma versão de fluxo no conjunto de dados inicial. Enviar um ID de arquivo de fluxo para um dispositivo pode simplificar a programação do firmware/software do dispositivo, pois elimina a necessidade de fazer um`DescribeStream`solicitação do dispositivo para obter esse ID. O dispositivo pode especificar a versão do stream em um`GetStream`solicitação para aplicar uma verificação de consistência caso o fluxo tenha sido atualizado inesperadamente.

Use DescribeStream para obter dados de fluxo

AWS IoTA entrega de arquivos baseada em MQTT fornece o`DescribeStream`API para enviar dados de fluxo para um dispositivo. Os dados do stream retornados por essa API incluem o ID do fluxo, a versão do

stream, a descrição do fluxo e uma lista de arquivos de fluxo, cada um com um ID de arquivo e o tamanho do arquivo em bytes. Com essas informações, um dispositivo pode selecionar arquivos arbitrários para iniciar o processo de transferência de dados.

> **Note**
>
> Não é necessário usar o`DescribeStream`API se o dispositivo receber todas as IDs de arquivo de fluxo necessárias no conjunto de dados inicial.

Siga estas etapas para criar um`DescribeStream`solicitação.

1. Assine o filtro de tópicos “aceito”`$aws/things/`*`ThingName`*`/streams/`*`StreamId`*`/description/json`.
2. Assine o filtro de tópicos “rejeitado”`$aws/things/`*`ThingName`*`/streams/`*`StreamId`*`/rejected/json`.
3. Publicar um`DescribeStream`solicitar enviando uma mensagem para`$aws/things/`*`ThingName`*`/streams/`*`StreamId`*`/describe/json`.
4. Se a solicitação foi aceita, seu dispositivo receberá um`DescribeStream`resposta no filtro de tópico “aceito”.
5. Se a solicitação foi rejeitada, o dispositivo receberá a resposta de erro no filtro de tópico “rejeitado”.

> **Note**
>
> Se você substituir`json`com`cbor`nos tópicos e filtros de tópicos mostrados, seu dispositivo recebe mensagens no formato CBOR, que é mais compacto que o JSON.

Solicitação DescribeStream

Um típico`DescribeStream`A solicitação no JSON é semelhante ao exemplo a seguir.

```
{
    "c": "ec944cfb-1e3c-49ac-97de-9dc4aaad0039"
}
```

- (Opcional)”c“é o campo token do cliente.

  O token do cliente não pode ter mais de 64 bytes. Um token de cliente com mais de 64 bytes resulta em uma resposta de erro e um`InvalidRequest`(Nenhum console do Windows encontrado. Você está executando o cmd.exe?”.

Resposta DescribeStream

UMA`DescribeStream`Veja no exemplo abaixo a aparência da resposta no JSON.

```
{
    "c": "ec944cfb-1e3c-49ac-97de-9dc4aaad0039",
    "s": 1,
    "d": "This is the description of stream ABC.",
    "r": [
        {
            "f": 0,
            "z": 131072
        },
        {
```

```
            "f": 1,
            "z": 51200
        }
    ]
}
```

- "`c`“é o campo token do cliente. Isso é retornado se for dado no`DescribeStream`solicitação. Use o token do cliente para associar a resposta à solicitação.
- "`s`“é a versão do stream como um número inteiro. Você pode usar esta versão para realizar uma verificação de consistência com seu`GetStream`Solicitações.
- "`r`“Contém uma lista dos arquivos no fluxo.
  - "`f`“é o ID do arquivo de fluxo como um inteiro.
  - "`z`“é o tamanho do arquivo de fluxo em número de bytes.
- "`d`“contém a descrição do fluxo.

Obter blocos de dados de um arquivo de fluxo

Você pode usar o`GetStream`API para que um dispositivo possa receber arquivos de fluxo em pequenos blocos de dados, para que ele possa ser usado por dispositivos que têm restrições ao processamento de grandes tamanhos de blocos. Para receber um arquivo de dados inteiro, um dispositivo pode precisar enviar ou receber várias solicitações e respostas até que todos os blocos de dados sejam recebidos e processados.

Solicitação GetStream

Siga estas etapas para criar um`GetStream`solicitação.

1. Assine o filtro de tópicos “aceito”`$aws/things/`*`ThingName`*`/streams/`*`StreamId`*`/data/json`.
2. Assine o filtro de tópicos “rejeitado”`$aws/things/`*`ThingName`*`/streams/`*`StreamId`*`/rejected/json`.
3. Publicar um`GetStream`solicitação para o tópico`$aws/things/`*`ThingName`*`/streams/`*`StreamId`*`/get/json`.
4. Se a solicitação for aceita, seu dispositivo receberá um ou mais`GetStream`respostas no filtro de tópico “aceito”. Cada mensagem de resposta contém informações básicas e uma carga útil de dados para um único bloco.
5. Repita as etapas 3 e 4 para receber todos os blocos de dados. Você deve repetir essas etapas se a quantidade de dados solicitada for maior que 128 KB. Você deve programar seu dispositivo para usar vários`GetStream`solicitações para receber todos os dados solicitados.
6. Se a solicitação foi rejeitada, seu dispositivo receberá a resposta de erro no filtro de tópico “rejeitado”.

   Note

   - Se você substituir “json” por “cbor” nos tópicos e filtros de tópicos mostrados, seu dispositivo receberá mensagens no formato CBOR, que é mais compacto que o JSON.
   - AWS IoTA entrega de arquivos baseada em MQTT limita o tamanho de um bloco a 128 KB. Se você fizer uma solicitação para um bloco com mais de 128 KB, a solicitação falhará.
   - Você pode fazer uma solicitação para vários blocos cujo tamanho total seja maior que 128 KB (por exemplo, se você fizer uma solicitação para 5 blocos de 32 KB cada para um total de 160 KB de dados). Nesse caso, a solicitação não falha, mas o dispositivo deve fazer várias solicitações para receber todos os dados solicitados. O serviço enviará blocos adicionais à medida que seu dispositivo fizer solicitações adicionais. Recomendamos que você continue

com uma nova solicitação somente após a resposta anterior ter sido recebida e processada corretamente.

- Independentemente do tamanho total dos dados solicitados, você deve programar seu dispositivo para iniciar novas tentativas quando os blocos não forem recebidos ou não recebidos corretamente.

Um típico`GetStream`A solicitação no JSON é semelhante ao exemplo a seguir.

```
{
    "c": "1bb8aaa1-5c18-4d21-80c2-0b44fee10380",
    "s": 1,
    "f": 0,
    "l": 4096,
    "o": 2,
    "n": 100,
    "b": "..."
}
```

- [opcional]”`c`“é o campo token do cliente.

  O token do cliente pode ter até 64 bytes. Um token de cliente com mais de 64 bytes resulta em uma resposta de erro e um`InvalidRequest`(Nenhum console do Windows encontrado. Você está executando o cmd.exe?”.
- [opcional]”`s`“é o campo de versão do fluxo (um número inteiro).

  A entrega de arquivos baseada em MQTT aplica uma verificação de consistência com base nessa versão solicitada e na versão mais recente do stream na nuvem. Se a versão do stream enviada de um dispositivo em um`GetStream`request não corresponde à versão mais recente do stream na nuvem, o serviço envia uma resposta de erro e um`VersionMismatch`(Nenhum console do Windows encontrado. Você está executando o cmd.exe?”. Normalmente, um dispositivo recebe a versão de fluxo esperada (mais recente) no conjunto de dados inicial ou na resposta a`DescribeStream`.
- "`f`“é o ID do arquivo de fluxo (um inteiro no intervalo de 0 a 255).

  O ID do arquivo de fluxo é necessário quando você cria ou atualiza um fluxo usando oAWS CLIou SDK. Se um dispositivo solicitar um arquivo de fluxo com um ID que não existe, o serviço enviará uma resposta de erro e um`ResourceNotFound`(Nenhum console do Windows encontrado. Você está executando o cmd.exe?”.
- "`l`“é o tamanho do bloco de dados em bytes (um número inteiro no intervalo de 256 a 131.072).

  ConsulteCrie um bitmap para uma solicitação GetStream (p. 878)para obter instruções sobre como usar os campos de bitmap para especificar qual parte do arquivo de fluxo será retornada na`GetStream`Resposta. Se um dispositivo especificar um tamanho de bloco que está fora do intervalo, o serviço enviará uma resposta de erro e um`BlockSizeOutOfBounds`(Nenhum console do Windows encontrado. Você está executando o cmd.exe?”.
- [opcional]”`o`“é o deslocamento do bloco no arquivo de fluxo (um inteiro no intervalo de 0 a 98.304).

  ConsulteCrie um bitmap para uma solicitação GetStream (p. 878)para obter instruções sobre como usar os campos de bitmap para especificar qual parte do arquivo de fluxo será retornada na`GetStream`Resposta. O valor máximo de 98.304 é baseado em um limite de tamanho de arquivo de fluxo de 24 MB e 256 bytes para o tamanho mínimo de bloco. O padrão é 0 se não for especificado.
- [opcional]”`n`“é o número de blocos solicitados (um número inteiro no intervalo de 0 a 98.304).

  O campo “n” especifica (1) o número de blocos solicitados ou (2) quando o campo bitmap (“b”) é usado, um limite no número de blocos que serão retornados pela solicitação de bitmap. Esse segundo uso é opcional. Se não for definido, o padrão é 131072/*`DatablockSize`*.
- [opcional]”`b`“é um bitmap que representa os blocos que estão sendo solicitados.

Usando um bitmap, seu dispositivo pode solicitar blocos não consecutivos, o que torna o tratamento de novas tentativas após um erro mais conveniente. ConsulteCrie um bitmap para uma solicitação GetStream (p. 878)para obter instruções sobre como usar os campos de bitmap para especificar qual parte do arquivo de fluxo será retornada na`GetStream`Resposta. Para este campo, converta o bitmap em uma string que representa o valor do bitmap em notação hexadecimal. O bitmap deve ser inferior a 12.288 bytes.

> **Important**
>
> Ou seja"`n`"ou"`b`"deve ser especificado. Se nenhum deles for especificado, o`GetStream`a solicitação pode não ser válida quando o tamanho do arquivo for menor que 131072 bytes (128 KB).

Resposta GetStream

UMA`GetStream`resposta em JSON se parece com este exemplo para cada bloco de dados solicitado.

```
{
    "c": "1bb8aaa1-5c18-4d21-80c2-0b44fee10380",
    "f": 0,
    "l": 4096,
    "i": 2,
    "p": "..."
}
```

- "`c`"é o campo token do cliente. Isso é retornado se for dado no`GetStream`solicitação. Use o token do cliente para associar a resposta à solicitação.
- "`f`"é o ID do arquivo de fluxo ao qual a carga útil do bloco de dados atual pertence.
- "`l`"é o tamanho da carga útil do bloco de dados em bytes.
- "`i`"é o ID do bloco de dados contido na carga útil. Os blocos de dados são numerados a partir de 0.
- "`p`"contém a carga útil do bloco de dados. Este campo é uma string, que representa o valor do bloco de dados emBase64Codificação.

Crie um bitmap para uma solicitação GetStream

É possível usar o campo bitmap (`b`) em um`GetStream`solicitação para obter blocos não consecutivos de um arquivo de fluxo. Isso ajuda dispositivos com capacidade de RAM limitada a lidar com problemas de entrega de rede. Um dispositivo pode solicitar somente os blocos que não foram recebidos ou não recebidos corretamente. O bitmap determina quais blocos do arquivo de fluxo serão retornados. Para cada bit, que é definido como 1 no bitmap, um bloco correspondente do arquivo de fluxo será retornado.

Aqui está um exemplo de como especificar um bitmap e seus campos de suporte em um`GetStream`solicitação. Por exemplo, você deseja receber um arquivo de fluxo em blocos de 256 bytes (o tamanho do bloco). Pense em cada bloco de 256 bytes como tendo um número que especifica sua posição no arquivo, a partir de 0. Portanto, o bloco 0 é o primeiro bloco de 256 bytes no arquivo, o bloco 1 é o segundo e assim por diante. Você deseja solicitar os blocos 20, 21, 24 e 43 do arquivo.

Bloquear offset

Como o primeiro bloco é o número 20, especifique o deslocamento (campo`o`) como 20 para economizar espaço no bitmap.

Número de blocos

Para garantir que seu dispositivo não receba mais blocos do que ele pode manipular com recursos de memória limitados, você pode especificar o número máximo de blocos que devem ser retornados

em cada mensagem enviada pela entrega de arquivos baseada em MQTT. Observe que esse valor é desconsiderado se o próprio bitmap especificar menos do que esse número de blocos, ou se ele tornaria o tamanho total das mensagens de resposta enviadas pela entrega de arquivos baseada em MQTT maior do que o limite de serviço de 128 KB por`GetStream`solicitação.

Bloquear bitmap

O bitmap em si é uma matriz de bytes não assinados expressos em notação hexadecimal e incluídos na`GetStream`Solicitação como uma representação de string do número. Mas, para construir essa string, vamos começar pensando no bitmap como uma longa sequência de bits (um número binário). Se um bit nessa sequência for definido como 1, o bloco correspondente do arquivo de fluxo será enviado de volta para o dispositivo. Para nosso exemplo, queremos receber os blocos 20, 21, 24 e 43, então devemos definir os bits 20, 21, 24 e 43 em nosso bitmap. Podemos usar o deslocamento de bloco para economizar espaço, portanto, depois de subtrair o deslocamento de cada número de bloco, queremos definir os bits 0, 1, 4 e 23, como o exemplo a seguir.

```
1 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
```

Tomando um byte (8 bits) de cada vez, isso é convencionalmente escrito como: “0b00010011", “0b00000000" e “0b10000000". O bit 0 aparece em nossa representação binária no final do primeiro byte e o bit 23 no início do último. Isso pode ser confuso, a menos que você conheça as convenções. O primeiro byte contém bits 7-0 (nessa ordem), o segundo byte contém bits 15-8, o terceiro byte contém bits 23-16 e assim por diante. Na notação hexadecimal, isso se converte em “0x130080".

Tip

Você pode converter o binário padrão em notação hexadecimal. Pegue quatro dígitos binários de cada vez e converta-os em seu equivalente hexadecimal. Por exemplo, “0001" se torna “1", “0011" se torna “3" e assim por diante.

Juntando tudo isso, o JSON para o nosso`GetStream`A solicitação é semelhante à seguinte.

```
{
    "c" : "1",       // client token
    "s" : 1,         // expected stream version
    "l" : 256,       // block size
```

```
    "f" : 1,         // source file index id
    "o" : 20,        // block offset
    "n" : 32,        // number of blocks
    "b" : "0x130080" // A missing blockId bitmap starting from the offset
}
```

Lidar com erros deAWS IoTEntrega de arquivos baseada em MQTT

Uma resposta de erro enviada para um dispositivo para ambos`DescribeStream`e`GetStream`As APIs contêm um token de cliente, um código de erro e uma mensagem de erro. Uma resposta de erro típica é semelhante ao exemplo a seguir.

```
{
    "o": "BlockSizeOutOfBounds",
    "m": "The block size is out of bounds",
    "c": "1bb8aaa1-5c18-4d21-80c2-0b44fee10380"
}
```

- "`o`“é o código de erro que indica o motivo pelo qual ocorreu um erro. Consulte os códigos de erro mais adiante nesta seção para obter mais detalhes.
- "`m`“é a mensagem de erro que contém detalhes do erro.
- "`c`“é o campo token do cliente. Isso pode ser retornado se for dado no`DescribeStream`solicitação. Você pode usar o token do cliente para associar a resposta à solicitação.

  O campo token do cliente nem sempre está incluído em uma resposta de erro. Quando o token do cliente fornecido na solicitação não é válido ou está mal formado, ele não é retornado na resposta de erro.

  > **Note**
  >
  > Para compatibilidade com versões anteriores, os campos na resposta de erro podem estar em formato não abreviado. Por exemplo, o código de erro pode ser designado pelos campos “código” ou “o” e o campo token do cliente pode ser designado pelos campos “ClientToken” ou “c”. Recomendamos que você use o formulário de abreviação mostrado acima.

InvalidTopic

O tópico MQTT da mensagem de fluxo é inválido.

InvalidJson

A solicitação Stream não é um documento JSON válido.

InvalidCBOR

A solicitação Stream não é um documento CBOR válido.

InvalidRequest

A solicitação geralmente é identificada como malformada. Para obter mais informações, consulte a mensagem de erro.

Não autorizado

A solicitação não está autorizada a acessar os arquivos de dados de stream no meio de armazenamento, como o Amazon S3. Para obter mais informações, consulte a mensagem de erro.

BlockSizeOutOfBounds

O tamanho do bloco está fora dos limites. Consulte o"Entrega de arquivos baseada em MQTT"seçãoAWS IoT CoreCotas de serviço do.

OffsetOutOfBounds

O deslocamento está fora dos limites. Consulte o"Entrega de arquivos baseada em MQTT"seçãoAWS IoT CoreCotas de serviço do.

BlockCountLimitExceeded

O número de blocos de solicitação está fora dos limites. Consulte o"Entrega de arquivos baseada em MQTT"seçãoAWS IoT CoreCotas de serviço do.

BlockBitMaplimitExceeded

O tamanho do bitmap de solicitação está fora dos limites. Consulte o"Entrega de arquivos baseada em MQTT"seçãoAWS IoT CoreCotas de serviço do.

ResourceNotFound

O fluxo solicitado, arquivos, versões de arquivos ou blocos não foram encontrados. Consulte a mensagem de erro para obter mais detalhes.

VersionMismatch

A versão do stream na solicitação não corresponde à versão do stream no recurso de entrega de arquivos baseado em MQTT. Isso indica que os dados do fluxo foram modificados desde que a versão do stream foi recebida inicialmente pelo dispositivo.

ETagmismatch

O S3 ETag no fluxo não corresponde à ETag da versão mais recente do objeto S3.

InternalError

Ocorreu um erro interno na entrega de arquivos baseada em MQTT.

Um exemplo de caso de uso no FreeRTOS OTA

O agente FreeRTOS OTA (over-the-air) usaAWS IoTEntrega de arquivos baseada em MQTT para transferir imagens de firmware FreeRTOS para dispositivos FreeRTOS. Para enviar o conjunto de dados inicial para um dispositivo, ele usa oAWS IoTServiço de Job para agendar um trabalho de atualização OTA para dispositivos FreeRTOS.

Para uma implementação de referência de um cliente de entrega de arquivos baseado em MQTT, consulteCódigos do agente FreeRTOS OTAna documentação do FreeRTOS.

AWS IoT Device Defender

O AWS IoT Device Defender é um serviço de segurança que permite auditar a configuração de seus dispositivos, monitorar dispositivos conectados para detectar comportamentos anormais e reduzir os riscos à segurança. Ele oferece a você a capacidade de aplicar políticas de segurança consistentes em toda a sua frota de dispositivos do AWS IoT e responder rapidamente quando os dispositivos estiverem comprometidos.

Frotas de IoT consistem em grandes quantidades de dispositivos com capacidades diversas, duradouros e geograficamente distribuídos. Essas características tornam a configuração da frota complexa e propensa a erros. E como os dispositivos, quase sempre, têm restrições quanto à capacidade computacional e aos recursos de memória e armazenamento, isso limita o uso de criptografia e outras formas de segurança nos próprios dispositivos. Além disso, muitas vezes, os dispositivos usam software com vulnerabilidades conhecidas. Esses fatores tornam frotas de IoT um alvo atrativo para hackers e tornam difícil proteger a frota de dispositivos de forma contínua.

AWS IoT Device DefenderO aborda esses desafios fornecendo ferramentas para identificar problemas de segurança e desvios das melhores práticas.AWS IoT Device DefenderPode auditar frotas de dispositivos para garantir que elas cumpram as melhores práticas de segurança e para detectar comportamentos anormais em dispositivos.

Treinamento e certificação da AWS

Faça o curso a seguir para começar a usarAWS IoT Device Defender:AWS IoT Device Defender Primer.

Conceitos básicos do AWS IoT Device Defender

Use os tutoriais a seguir para trabalhar com o AWS IoT Device Defender.

Tópicos

- Configuração (p. 882)
- Guia de auditoria (p. 884)
- Guia de detecção de ML (p. 896)
- Personalize quando e como você visualizaAWS IoT Device DefenderAuditoria de resultados (p. 919)

Configuração

Antes de usar o AWS IoT Device Defender pela primeira vez, execute as seguintes tarefas:

- Cadastre-se no AWS (p. 883)
- Criar um usuário do IAM (p. 883)

Essas tarefas criam umConta da AWSe um usuário do IAM com privilégios do administrador da conta.

Cadastre-se no AWS

Ao cadastrar-se na AWS, sua conta é automaticamente cadastrada em todos os serviços da AWS incluindo o AWS IoT Device Defender. Se você já tem uma Conta da AWS, passe para a próxima tarefa. Se você não tem uma Conta da AWS, siga o procedimento abaixo para criar uma.

Se você ainda não tem Conta da AWS, siga as etapas a seguir para criar um.

Para se cadastrar em uma Conta da AWS

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

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

Anote o número da sua Conta da AWS, pois você precisará dele na próxima tarefa.

Criar um usuário do IAM

Este procedimento descreve como criar um usuário do IAM para você e adicionar esse usuário a um grupo com permissões administrativas de uma política gerenciada anexada.

Para criar um usuário administrador para você mesmo e adicionar o usuário a um grupo de administradores (console)

1. Faça login no console do IAM como proprietário da conta escolhendo Root user (Usuário root) e inserindo o endereço de e-mail da sua Conta da AWS. Na próxima página, insira sua senha.

   Note

   Recomendamos seguir as práticas recomendadas para utilizar o usuário do IAM **`Administrator`** a seguir e armazenar as credenciais do usuário raiz com segurança. Cadastre-se como o usuário raiz apenas para executar algumas tarefas de gerenciamento de serviços e contas.
2. No painel de navegação, escolha Users (Usuários) e Add users (Adicionar usuários).
3. Em User name (Nome do usuário), digite **`Administrator`**.
4. Marque a caixa de seleção ao lado do acesso ao AWS Management Console. Então, selecione Custom password (Senha personalizada), e insira sua nova senha na caixa de texto.
5. (Opcional) Por padrão, a AWS exige que o novo usuário crie uma senha ao fazer login pela primeira vez. Você pode desmarcar a caixa de seleção próxima de User must create a new password at next sign-in (O usuário deve criar uma senha no próximo login) para permitir que o novo usuário redefina a senha depois de fazer login.
6. Selecione Next (Próximo): Permissions
7. Em Set permissions (Conceder permissões), escolha Add user to group (Adicionar usuário ao grupo).
8. Escolha Create group (Criar grupo).
9. Na caixa de diálogo Create group (Criar grupo), em Group name (Nome do grupo), digite **`Administrators`**.
10. Escolha Filter policies (Filtrar políticas) e, depois, selecione AWS managed — job function (Função de trabalho gerenciada da AWS) para filtrar o conteúdo da tabela.
11. Na lista de políticas, marque a caixa de seleção AdministratorAccess. A seguir escolha Criar grupo.

    Note

    Você deve ativar o acesso de usuário do IAM e da função para Billing (Faturamento) antes de usar as permissões de `AdministratorAccess` para acessar o console do AWS Billing

and Cost Management. Para fazer isso, siga as instruções na etapa 1 do tutorial sobre como delegar acesso ao console de faturamento.

12. Suporte a lista de grupos, selecione a caixa de seleção para seu novo grupo. Escolha Refresh (Atualizar) caso necessário, para ver o grupo na lista.
13. Selecione Next (Próximo): Tags.
14. (Opcional) Adicione metadados ao usuário anexando tags como pares de chave-valor. Para obter mais informações sobre como usar tags no IAM, consulte Marcar entidades do IAM no Guia do usuário do IAM.
15. Selecione Next (Próximo): Review (Revisar)Para ver uma lista de associações a grupos a serem adicionadas ao novo usuário. Quando você estiver pronto para continuar, escolha Create user (Criar usuário).

Você pode usar esse mesmo processo para criar mais grupos e usuários e conceder aos seus usuários acesso aos recursos da sua Conta da AWS. Para saber como usar políticas para restringir as permissões de usuário a recursos específicos da AWS, consulte Gerenciamento de acesso e Exemplos de políticas.

Guia de auditoria

Este tutorial fornece instruções sobre como configurar uma auditoria recorrente, configurar alarmes, revisar resultados de auditoria e mitigar problemas de auditoria.

Tópicos

- Pré-requisitos (p. 884)
- Permitir verificações da auditoria (p. 884)
- Visualizar resultados da auditoria (p. 888)
- Criação de ações de mitigação de auditoria (p. 889)
- Aplique ações de mitigação às descobertas de auditoria (p. 891)
- Ativar notificações do SNS (opcional) (p. 892)
- Habilitar log (opcional) (p. 894)
- Criar umAWS IoT Device DefenderFunção do IAM de auditoria (opcional) (p. 895)

Pré-requisitos

Para concluir este tutorial, você precisará do seguinte:

- Uma Conta da AWS. Se você não tiver isso, consulteConfigurar o.

Permitir verificações da auditoria

No procedimento a seguir, você habilita verificações de auditoria que analisam configurações e políticas de contas e dispositivos para garantir que as medidas de segurança estejam em vigor. Neste tutorial, instruímos você a ativar todas as verificações de auditoria, mas você pode selecionar as verificações que desejar.

O preço da auditoria é por contagem de dispositivos por mês (dispositivos de frota conectados aAWS IoT). Portanto, adicionar ou remover verificações de auditoria não afetaria sua fatura mensal ao usar esse recurso.

1. NoAWS IoTconsoleNo painel de navegação, expandaDefendere selecioneConceitos básicos do.

2. OConceitos básicos do Device Defender Audita tela fornece uma visão geral das etapas necessárias para habilitar as verificações de auditoria. Depois de revisar a tela, selecionePróximo.
3. Se já tiver uma função a ser usada, selecione. Caso contrário, selecioneCriar função doe nomeie`AWSIoTDeviceDefenderAudit`.

Você verá as permissões necessárias anexadas automaticamente à função. Selecione os triângulos ao lado dePermissõeseRelações de confiançapara ver quais permissões são concedidas. SelectPróximoQuando você estiver pronto para seguir em frente.

4. NoSelecione verificaçõesvocê verá todas as verificações de auditoria que você pode selecionar. Para este tutorial, instruímos você a selecionar todas as verificações, mas você pode selecionar as verificações desejadas. Ao lado de cada verificação de auditoria, há um ícone de ajuda que descreve o que a verificação de auditoria faz. Para obter mais informações sobre verificações de auditoria, consulteVerificações da auditoria.

   SelectPróximodepois de selecionar seus cheques.

Você poderá alterar as verificações de auditoria configuradas emConfigurações.

5. NoConfigurar o SNS (opcional)tela, selecioneHabilitar auditoria. Se quiser ativar notificações do SNS, consulteAtivar notificações do SNS (opcional) (p. 892).

6. Você será redirecionado paraSchedulesembaixoAuditoria.

Visualizar resultados da auditoria

O procedimento a seguir mostra como visualizar os resultados da auditoria. Neste tutorial, você verá os resultados da auditoria das verificações de auditoria configuradas noPermitir verificações da auditoria (p. 884)para "Hello, World!".

Para visualizar os resultados da auditoria

1. NoAWS IoTconsoleNo painel de navegação, expandaDefender, selecioneAuditoriae selecioneResultados.
2. OSummary (Resumo)informará se você tem verificações não compatíveis.

3. Selecione oName (Nome)da verificação de auditoria que você gostaria de investigar.

4. Use os pontos de interrogação para obter orientações sobre como tornar suas verificações não compatíveis. Por exemplo, você pode seguirHabilitar log (opcional) (p. 894)para tornar a verificação “Registro desativado” compatível.

Criação de ações de mitigação de auditoria

No procedimento a seguir, você criará umAWS IoT Device DefenderAção de mitigação de auditoria para habilitarAWS IoTregistro em log. Cada verificação de auditoria mapeou ações de mitigação que afetarão quaisTipo de açãovocê escolhe para a verificação de auditoria que deseja corrigir. Para obter mais informações, consulteAções de mitigação.

Para usar o console do AWS IoT para criar ações de mitigação.

1. Abra o console do AWS IoT.
2. No painel de navegação à esquerda, escolha Defend (Defender) e escolha Mitigation Actions (Ações de mitigação).
3. Na página Mitigation Actions (Ações de mitigação), escolha Create (Criar).

4. NoCriar uma ação de mitigaçãoPágina, emAction name (Nome da ação), insira um nome exclusivo para sua ação de mitigação, como*`EnableErrorLoggingAction`*.
5. DentroTipo de ação, escolhaHabilitar registro em log de IoT.
6. DentroFunção de execução de ação, selecioneCriar função do. para oName (Nome), use*`IOTMitigationActionErrorLoggingRole`*. Em seguida, escolhaCriar função do.
7. DentroParâmetros, UNDERFunção do registro em log, selecione`AWSIoTLoggingRole`. para oNível de log, escolha`Error`.

8. Escolha Save (Salvar) para salvar a ação de mitigação em sua conta da AWS.

9. Uma vez criado, você verá a seguinte indicação de tela de que sua ação de mitigação foi criada com sucesso.

Aplique ações de mitigação às descobertas de auditoria

O procedimento a seguir mostra como aplicar ações de mitigação aos resultados da auditoria.

Para mitigar descobertas de auditoria não compatíveis

1. Abra o console do AWS IoT.
2. No painel de navegação à esquerda, escolhaAuditoriae, depois, escolhaResultados. Selecione o nome da auditoria à qual você deseja responder.
3. Verifique seus resultados. Observe que`Logging disabled`está localizado emVerificações incompatíveis.
4. SelectIniciar ações de mitigação.

5. UNDERSelecione ações do, selecione as ações apropriadas para cada descoberta não compatível para resolver os problemas.

6. SelectConfirmar.
7. Depois que a ação de mitigação for iniciada, pode levar alguns minutos para que ela seja executada.

Para verificar se a ação de mitigação funcionou

1. NoAWSNo console do, no painel de navegação, selecioneConfigurações.
2. Confirmar queLogsestá`Enabled`O e aNível de verbosidadeé`Error`.

Ativar notificações do SNS (opcional)

No procedimento a seguir, você habilita as notificações do Serviço de Notificações Simples (SNS) para alertá-lo quando suas auditorias identificarem recursos não compatíveis. Neste tutorial, você configurará notificações para as verificações de auditoria ativadas noPermitir verificações da auditoria (p. 884)para "Hello, World!".

1. Primeiro, você precisa criar uma política do IAM que conceda acesso ao Amazon SNS por meio daAWSConsole de gerenciamento. Para isso, você pode seguir oCriar umAWS IoT Device DefenderFunção do IAM de auditoria (opcional) (p. 895)processo, mas selecionandoAWSIoTDeviceDefenderPublishFindingsToSNSMitigationActionna etapa 8.
2. NoAWS IoTconsoleNo painel de navegação, expandaDefendere selecioneConfigurações.

3. UNDERAlertas do SNS, selecioneEdite.

4. NoEditar alertas do SNStela, selecioneEnabled (Habilitado). UNDERTópico, selecioneCriar. Nomeie o tópico`Notificações IOTDD`e selecioneCriar. UNDERFunção, selecione a função criada chamada`AWSIoTDeviceDefenderAudit`.

Select Update (Atualizar).

Se você quiser receber e-mail ou texto em suas plataformas de operações por meio do SNS, consulteUsar o Amazon SNS para notificações de usuários.

Habilitar log (opcional)

Este procedimento descreve como ativarAWS IoTPara registrar informações no CloudWatch Logs. Isso permitirá que você visualize os resultados da auditoria. Habilitar o registro em log pode resultar em cobranças incorridas.

Para habilitar o registro em log

1. NoAWSNo console do, no painel de navegação, selecioneConfigurações.
2. UNDERLogs, selecioneEdite.

3. UNDERNível de verbosidade, selecioneDepuração (mais detalhada).
4. UNDERDefinir função, selecioneCriar função doe nomeie a função`AWSIOTLoggingRole`. Uma política será anexada automaticamente.

Select Update (Atualizar).

Criar umAWS IoT Device DefenderFunção do IAM de auditoria (opcional)

No procedimento a seguir, crie umaAWS IoT Device DefenderFunção do IAM de auditoria que forneceAWS IoT Device DefenderAcesso de leitura aoAWS IoT.

1. Navegue até o console do IAM emhttps://console.aws.amazon.com/iam/
2. No painel de navegação, escolhaUsuários doe, depois, escolhaAdicionar usuário.
3. Em User name (Nome do usuário), digite **`Administrator`**.
4. Marque a caixa de seleção ao lado daAWSAcesso ao Console de Gerenciamento. Então, selecione Custom password (Senha personalizada), e insira sua nova senha na caixa de texto.

5. (Opcional) Por padrão, a AWS exige que o novo usuário crie uma senha ao fazer login pela primeira vez. Você pode desmarcar a caixa de seleção próxima de User must create a new password at next sign-in (O usuário deve criar uma senha no próximo login) para permitir que o novo usuário redefina a senha depois de fazer login.
6. Selecione Next (Próximo): Permissions
7. Em Set permissions (Definir permissões), selecione Attach existing policies directly (Anexar políticas existentes diretamente).
8. Na lista de políticas, marque a caixa de seleção paraAWSIoTDeviceDefenderAudit.
9. Selecione Next (Próximo): Tags.
10. Selecione Next (Próximo): Review (Revisar)Para ver a lista de associações a grupos a serem adicionadas ao novo usuário. Quando você estiver pronto para continuar, selecione Criar usuário.

Guia de detecção de ML

Neste guia de conceitos básicos, você cria um Perfil de Segurança de Detecção de ML que usa aprendizado de máquina (ML) para criar modelos de comportamento esperado com base em dados de métricas históricas de seus dispositivos. Enquanto o ML Detect está criando o modelo de ML, você poderá monitorar o progresso do. Depois que o modelo de ML for criado, você poderá visualizar e investigar alarmes continuamente e mitigar problemas identificados.

Para obter mais informações sobre o ML Detect e seus comandos API e CLI, consulteDetecção de ML (p. 1003).

Este capítulo contém as seguintes seções:

- Pré-requisitos (p. 896)
- Como usar o ML Detect no console (p. 896)
- Como usar o ML Detect com a CLI (p. 910)

Pré-requisitos

- Uma Conta da AWS. Se você não tiver isso, consulteConfigurar o.

Como usar o ML Detect no console

Tutoriais

- Enable ML Detect (p. 896)
- Monitore o status do modelo de ML (p. 901)
- Revise seus alarmes ML Detect (p. 902)
- Ajuste seus alarmes de ML (p. 904)
- Marque o estado de verificação do alarme (p. 905)
- Mitigar problemas de dispositivos identificados (p. 906)

Enable ML Detect

Os procedimentos a seguir detalham como configurar o ML Detect no console do.

1. Primeiro, certifique-se de que seus dispositivos criarão os pontos de dados mínimos necessários, conforme definido emRequisitos mínimos ML Detect (p. 1004)para treinamento contínuo e atualização

do modelo. Para que a coleta de dados progrida, certifique-se de que seu Perfil de segurança esteja anexado a um alvo, que pode ser uma coisa ou grupo de coisas.

2. NoAWS IoTconsole, no painel de navegação, expandaDefender. SelecioneDetectar,Perfis de segurança,Criar perfil de segurança, e depoisCriar anomalia de ML Detectar perfil.
3. NoDefinir configurações básicasPage (página), faça o seguinte:
   - UNDERDestino, escolha seus grupos de dispositivos de destino.
   - UNDERNome do perfil de segurança, insira um nome para o Perfil de segurança.
   - (Opcional) EmDescrição Você pode escrever em uma breve descrição para o perfil de ML.
   - UNDERComportamentos de métricas selecionados no perfil de segurança, escolha as métricas que você gostaria de monitorar.

Quando concluir, selecione Próximo.

4. NoDefinir SNS (opcional), especifique um tópico do SNS para notificações de alarme quando um dispositivo viola um comportamento em seu perfil. Escolha uma função do IAM usada para publicar no tópico do SNS selecionado.

Se você ainda não tiver uma função do SNS, use as etapas a seguir para criar uma função com as permissões adequadas e relacionamentos de confiança necessários.

- Navegue até o .Console do IAM. No painel de navegação, escolha Roles e Create role.
- UNDERSelecione o tipo de entidade confiável, selecioneAWSServiço. Então, emEscolha um caso de uso, escolhaIoTe, emSelecione seu caso de uso, escolhaIoT - Ações de mitigação do Device Defender. Quando tiver terminado, escolhaPróximo: Permissions
- UNDERPolíticas de permissão anexadas, assegure-se de queAWSIoTDeviceDefenderPublishFindingsToSNSMitigationActionÉ selecionado e depois escolhaPróximo: Tags.

- UNDERAdicionar tags (opcional), você pode adicionar todas as tags que deseja associar à função. Quando tiver terminado, escolhaPróximo: Análise.
- UNDERReview (Revisar), dê um nome à sua função e garanta queAWSIoTDeviceDefenderPublishFindingsToSNSMitigationActionestá listado emPermissõeseAWSserviço: iot.amazonaws.comestá listado emRelações de confiança. Quando tiver terminado, escolhaCriar função do.

5. NoEditar comportamento de métrica, você pode personalizar suas configurações de comportamento de ML.

6. Quando concluir, selecione Próximo.
7. NoAnalisar a configuração, verifique os comportamentos que você gostaria de monitorar o aprendizado de máquina e depois escolhaPróximo.

8. Depois de criar o perfil de segurança, você será redirecionado para aPerfis de segurança, onde o Perfil de segurança recém-criado aparece.

   Note

   O treinamento e a criação inicial do modelo ML leva 14 dias para ser concluído. Você pode esperar ver alarmes após a conclusão, se houver alguma atividade anômala em seus dispositivos.

Monitore o status do modelo de ML

Enquanto seus modelos de ML estão no período de treinamento inicial, você pode monitorar o progresso deles a qualquer momento seguindo as etapas a seguir.

1. NoAWS IoTconsole, no painel de navegação, expandaDefendere, depois, escolhaDetectar,Perfis de segurança.
2. NoPerfis de segurança, escolha o Perfil de segurança que você gostaria de revisar. Em seguida, escolhaComportamentos e treinamento em ML.
3. NoComportamentos e treinamento em ML, verifique o andamento do treinamento de seus modelos de ML.

   Depois que o status do modelo forAtivo, ele começará a tomar decisões Detectar com base em seu uso e atualizará o perfil todos os dias.

Note

Se o modelo não progredir conforme o esperado, verifique se os dispositivos estão atendendo aoRequisitos mínimos (p. 1004).

Revise seus alarmes ML Detect

Depois que seus modelos de ML estiverem construídos e prontos para inferência de dados, você poderá visualizar e investigar regularmente alarmes identificados pelos modelos.

1. No AWS IoT console, no painel de navegação, expanda Defender e, depois, escolha Detectar, Alarmes.

2. Se você navegar até o Histórico, você também pode visualizar detalhes sobre seus dispositivos que não estão mais em alarmes.

Para obter mais informações, emGerencieescolherCoisas, escolha a coisa para a qual você gostaria de ver mais detalhes e, em seguida, navegue atéMétricas Defender. Você pode acessar oGráfico de métricas Defendere realize sua investigação sobre qualquer coisa em alarme doAtivoGuia. Nesse caso, o gráfico mostra um pico no tamanho da mensagem, que acionou o alarme. Você pode ver o alarme posteriormente apagado.

Ajuste seus alarmes de ML

Depois que seus modelos de ML estiverem criados e prontos para avaliações de dados, você poderá atualizar as configurações de comportamento de ML do Perfil de Segurança para alterar a configuração. O procedimento a seguir mostra como atualizar as configurações de comportamento de ML do Perfil de Segurança naAWS CLI.

1. NoAWS IoTconsole, no painel de navegação, expandaDefendere, depois, escolhaDetectar,Perfis de segurança.
2. NoPerfis de segurança, marque a caixa de seleção ao lado do Perfil de segurança que você deseja revisar. Em seguida, escolhaAções,Edite.

3. UNDERDefinir configurações básicas, você pode ajustar grupos de coisas de destino do Perfil de Segurança ou alterar as métricas que deseja monitorar.

4. Você pode atualizar qualquer uma das seguintes ações navegando paraEditar comportamentos métricos.
   - Os datapoints do modelo ML necessários para acionar o alarme
   - Seus pontos de dados do modelo ML necessários para limpar o alarme
   - Seu nível de confiança ML Detect
   - Suas notificações de Detecção de ML (por exemplo,Não suprimido,Suprimido)

Marque o estado de verificação do alarme

Marque seus alarmes definindo o estado de verificação e fornecendo uma descrição desse estado de verificação. Isso ajuda você e sua equipe a identificar alarmes aos quais você não precisa responder.

1. NoAWS IoTconsole, no painel de navegação, expandaDefendere, depois, escolhaDetectar,Alarmes. Selecione um alarme para marcar seu estado de verificação.

2. SelecioneMarcar estado de verificação. O modal de estado de verificação é aberto.
3. Escolha o estado de verificação apropriado, insira uma descrição de verificação (opcional) e, em seguida, escolhaMark. Essa ação atribui um estado de verificação e uma descrição ao alarme escolhido.

Mitigar problemas de dispositivos identificados

1. (Optional)Antes de configurar ações de mitigação de quarentena, vamos configurar um grupo de quarentena para onde moveremos o dispositivo que está violando. Você também pode usar um grupo existente.
2. Navegue atéGerencie,Grupos de coisas, e depoisCriar grupo de coisas. Dê um nome ao seu grupo de coisas. Para este tutorial, nomearemos nosso grupo de coisas`Quarantine_group`. UNDERGrupo de coisas,Segurança, aplique a política a seguir ao grupo de coisas.

```
{
  "Version": "2012-10-17",
```

```
  "Statement": [
    {
      "Effect": "Deny",
      "Action": "iot:*",
      "Resource": "*",
    }
  ]
}
```

Quando tiver terminado, escolhaCriar grupo de coisas.

3. Agora que criamos um grupo de coisas, vamos criar uma ação de mitigação que mova dispositivos que estão em alarme para o`Quarantine_group`.

   UNDERDefender,Ações de mitigação, escolhaCriar.

4. NoCriar uma nova ação de mitigação, insira as seguintes informações.

   - Action name (Nome da ação): Dê um nome à ação de mitigação, como**`Quarantine_action`**.
   - Tipo de ação: Escolha o tipo de ação. EscolheremosAdicionar coisas ao grupo de coisas (mitigação de auditoria ou detecção).
   - Função de execução da ação: Crie uma função ou escolha uma função existente se você tiver criado uma anteriormente.
   - Parâmetros: Escolha um grupo de coisas. Podemos usar`Quarantine_group`, que criamos anteriormente.

Quando você terminar, selecione Salvar. Agora você tem uma ação de mitigação que move dispositivos em alarme para um grupo de coisas de quarentena e uma ação de mitigação para isolar o dispositivo enquanto você investiga.

5. Navegue atéDefender,Detectar,Alarmes. Você pode ver quais dispositivos estão no estado de alarme emAtivo.

Selecione o dispositivo que você deseja mover para o grupo de quarentena e escolhaIniciar ações de mitigação.

6. UNDERIniciar ações de mitigação,Iniciar açõesSelecione a ação de mitigação criada anteriormente. Por exemplo, escolheremos**`Quarantine_action`**Depois, escolhaIniciar. A página Tarefas de ação é aberta.

7. O dispositivo está isolado em**`Quarantine_group`**E você pode investigar a causa raiz do problema que acionou o alarme. Depois de concluir a investigação, você pode mover o dispositivo para fora do grupo de coisas ou tomar outras ações.

Como usar o ML Detect com a CLI

A seguir mostra como configurar o ML Detect usando a CLI da

Tutoriais

- Enable ML Detect (p. 910)
- Monitore o status do modelo de ML (p. 911)
- Revise seus alarmes ML Detect (p. 913)
- Ajuste seus alarmes de ML (p. 914)
- Marque o estado de verificação do alarme (p. 916)
- Mitigar problemas de dispositivos identificados (p. 916)

Enable ML Detect

O procedimento a seguir mostra como ativar a Detectação de ML naAWS CLI.

1. Certifique-se de que seus dispositivos criarão os pontos de dados mínimos necessários, conforme definido emRequisitos mínimos ML Detect (p. 1004)para treinamento contínuo e atualização do modelo. Para que a coleta de dados progrida, certifique-se de que suas coisas estejam em um grupo de coisas anexado a um Perfil de segurança.
2. Crie um perfil de segurança de detecção de ML usando o`create-security-profile`comando. O exemplo a seguir cria um perfil de segurança chamado*`perfil de segurança para luzes inteligentes`*que verifica o número de mensagens enviadas, o número de falhas de autorização, o número de tentativas de conexão e o número de desconexões. O exemplo usa`mlDetectionConfig`para estabelecer que a métrica usará o modelo ML Detect.

```
aws iot create-security-profile \
    --security-profile-name security-profile-for-smart-lights \
    --behaviors \
     '[{
    "name": "num-messages-sent-ml-behavior",
    "metric": "aws:num-messages-sent",
    "criteria": {
      "consecutiveDatapointsToAlarm": 1,
      "consecutiveDatapointsToClear": 1,
      "mlDetectionConfig": {
        "confidenceLevel": "HIGH"
      }
    },
    "suppressAlerts": true
  },
  {
    "name": "num-authorization-failures-ml-behavior",
    "metric": "aws:num-authorization-failures",
    "criteria": {
      "consecutiveDatapointsToAlarm": 1,
      "consecutiveDatapointsToClear": 1,
```

```
      "mlDetectionConfig": {
        "confidenceLevel": "HIGH"
      }
    },
    "suppressAlerts": true
  },
  {
    "name": "num-connection-attempts-ml-behavior",
    "metric": "aws:num-connection-attempts",
    "criteria": {
      "consecutiveDatapointsToAlarm": 1,
      "consecutiveDatapointsToClear": 1,
      "mlDetectionConfig": {
        "confidenceLevel": "HIGH"
      }
    },
    "suppressAlerts": true
  },
  {
    "name": "num-disconnects-ml-behavior",
    "metric": "aws:num-disconnects",
    "criteria": {
      "consecutiveDatapointsToAlarm": 1,
      "consecutiveDatapointsToClear": 1,
      "mlDetectionConfig": {
        "confidenceLevel": "HIGH"
      }
    },
    "suppressAlerts": true
  }]'
```

Resultado:

```
{
    "securityProfileName": "security-profile-for-smart-lights",
    "securityProfileArn": "arn:aws:iot:eu-west-1:123456789012:securityprofile/security-
profile-for-smart-lights"
  }
```

3. Em seguida, associe seu Perfil de segurança a um ou vários grupos de coisas. Usar a`attach-security-profile`Para anexar um grupo de coisas ao perfil de segurança. O exemplo a seguir associa um grupo de coisas chamado*ML_DETECT_BETA_STATIC_Group*com o*perfil de segurança para luzes inteligentes*Perfil de segurança.

```
aws iot attach-security-profile \
--security-profile-name security-profile-for-smart-lights \
--security-profile-target-arn arn:aws:iot:eu-
west-1:123456789012:thinggroup/ML_Detect_beta_static_group
```

Resultado:

Nenhum.

4. Depois de criar seu perfil de segurança completo, o modelo de ML inicia o treinamento. O treinamento e a construção inicial do modelo ML leva 14 dias para ser concluído. Após 14 dias, se houver atividade anômala no seu dispositivo, você pode esperar ver alarmes.

Monitore o status do modelo de ML

O procedimento a seguir mostra como monitorar o treinamento em andamento dos modelos de ML.

- Usar a`get-behavior-model-training-summaries`comando para visualizar o progresso do modelo de ML. O exemplo a seguir obtém o resumo do progresso do treinamento do modelo de ML para o*perfil de segurança para luzes inteligentes*Perfil de segurança.`modelStatus`mostra se um modelo concluiu o treinamento ou ainda está pendente de compilação para um determinado comportamento.

```
aws iot get-behavior-model-training-summaries \
   --security-profile-name security-profile-for-smart-lights
```

Resultado:

```
{
    "summaries": [
        {
            "securityProfileName": "security-profile-for-smart-lights",
            "behaviorName": "Messages_sent_ML_behavior",
            "trainingDataCollectionStartDate": "2020-11-30T14:00:00-08:00",
            "modelStatus": "ACTIVE",
            "datapointsCollectionPercentage": 29.408,
            "lastModelRefreshDate": "2020-12-07T14:35:19.237000-08:00"
        },
        {
            "securityProfileName": "security-profile-for-smart-lights",
            "behaviorName": "Messages_received_ML_behavior",
            "modelStatus": "PENDING_BUILD",
            "datapointsCollectionPercentage": 0.0
        },
        {
            "securityProfileName": "security-profile-for-smart-lights",
            "behaviorName": "Authorization_failures_ML_behavior",
            "trainingDataCollectionStartDate": "2020-11-30T14:00:00-08:00",
            "modelStatus": "ACTIVE",
            "datapointsCollectionPercentage": 35.464,
            "lastModelRefreshDate": "2020-12-07T14:29:44.396000-08:00"
        },
        {
            "securityProfileName": "security-profile-for-smart-lights",
            "behaviorName": "Message_size_ML_behavior",
            "trainingDataCollectionStartDate": "2020-11-30T14:00:00-08:00",
            "modelStatus": "ACTIVE",
            "datapointsCollectionPercentage": 29.332,
            "lastModelRefreshDate": "2020-12-07T14:30:44.113000-08:00"
        },
        {
            "securityProfileName": "security-profile-for-smart-lights",
            "behaviorName": "Connection_attempts_ML_behavior",
            "trainingDataCollectionStartDate": "2020-11-30T14:00:00-08:00",
            "modelStatus": "ACTIVE",
            "datapointsCollectionPercentage": 32.891999999999996,
            "lastModelRefreshDate": "2020-12-07T14:29:43.121000-08:00"
        },
        {
            "securityProfileName": "security-profile-for-smart-lights",
            "behaviorName": "Disconnects_ML_behavior",
            "trainingDataCollectionStartDate": "2020-11-30T14:00:00-08:00",
            "modelStatus": "ACTIVE",
            "datapointsCollectionPercentage": 35.46,
            "lastModelRefreshDate": "2020-12-07T14:29:55.556000-08:00"
        }
    ]
}
```

Note

Se o modelo não progredir conforme o esperado, verifique se os dispositivos estão atendendo aoRequisitos mínimos (p. 1004).

Revise seus alarmes ML Detect

Depois que seus modelos de ML estiverem construídos e prontos para avaliações de dados, você poderá visualizar regularmente todos os alarmes inferidos pelos modelos. O procedimento a seguir mostra como visualizar seus alarmes naAWS CLI.

- Para ver todos os alarmes ativos, use o`list-active-violations`comando.

```
aws iot list-active-violations \
--max-results 2
```

Resultado:

```
{
    "activeViolations": []
}
```

Como alternativa, você pode visualizar todas as violações descobertas durante um determinado período usando o`list-violation-events`comando. O exemplo a seguir lista eventos de violação de 22 de setembro de 2020 5:42:13 GMT a 26 de outubro de 2020 5:42:13 GMT.

```
aws iot list-violation-events \
    --start-time 1599500533 \
    --end-time 1600796533 \
    --max-results 2
```

Resultado:

```
{
    "violationEvents": [
        {
            "violationId": "1448be98c09c3d4ab7cb9b6f3ece65d6",
            "thingName": "lightbulb-1",
            "securityProfileName": "security-profile-for-smart-lights",
            "behavior": {
                "name": "LowConfidence_MladBehavior_MessagesSent",
                "metric": "aws:num-messages-sent",
                "criteria": {
                    "consecutiveDatapointsToAlarm": 1,
                    "consecutiveDatapointsToClear": 1,
                    "mlDetectionConfig": {
                        "confidenceLevel": "HIGH"
                    }
                },
                "suppressAlerts": true
            },
            "violationEventType": "alarm-invalidated",
            "violationEventTime": 1600780245.29
        },
        {
            "violationId": "df4537569ef23efb1c029a433ae84b52",
            "thingName": "lightbulb-2",
            "securityProfileName": "security-profile-for-smart-lights",
            "behavior": {
                "name": "LowConfidence_MladBehavior_MessagesSent",
```

```
                "metric": "aws:num-messages-sent",
                "criteria": {
                    "consecutiveDatapointsToAlarm": 1,
                    "consecutiveDatapointsToClear": 1,
                    "mlDetectionConfig": {
                        "confidenceLevel": "HIGH"
                    }
                },
                "suppressAlerts": true
            },
            "violationEventType": "alarm-invalidated",
            "violationEventTime": 1600780245.281
        }
    ],
    "nextToken":
 "Amo6XIUrsOohsojuIG6TuwSR3X9iUvH2OCksBZg6bed2j21VSnD1uP1pflxKX1+a3cvBRSosIB0xFv40kM6RYBknZ/
vxabMe/ZW31Ps/WiZHlr9Wg7R7eEGli59IJ/U0iBQ1McP/ht0E2XA2TTIvYeMmKQQPsRj/
eoV9j7P/wveu7skNGepU/mvpV0O2Ap7hnV5U+Prx/9+iJA/341va
+pQww7jpUeHmJN9Hw4MqW0ysw0Ry3w38hOQWEpz2xwFWAxAARxeIxCxt5c37RK/lRZBlhYqoB
+w2PZ74730h8pICGY4gktJxkwHyyRabpSM/G/f5DFrD9O5v8idkTZzBxW2jrbzSUIdafPtsZHL/
yAMKr3HAKtaABz2nTsOBNre7X2d/jIjjarhon0Dh9l+8I9Y5Ey
+DIFBcqFTvhibKAafQt3gs6CUiqHdWiCenfJyb8whmDE2qxvdxGElGmRb
+k6kuN5jrZxxw95gzfYDgRHv11iEn8h1qZLD0czkIFBpMppHj9cetHPvM
+qffXGAzKi8tL6eQuCdMLXmVE3jbqcJcjk9ItnaYJi5zKDz9FVbrz9qZZPtZJFHp"
}
```

Ajuste seus alarmes de ML

Depois que seus modelos de ML estiverem criados e prontos para avaliações de dados, você poderá atualizar as configurações de comportamento de ML do Perfil de Segurança para alterar a configuração. O procedimento a seguir mostra como atualizar as configurações de comportamento de ML do Perfil de Segurança naAWS CLI.

- Para alterar as configurações de comportamento de ML do seu perfil de segurança, use o`update-security-profile`comando. O exemplo a seguir atualiza a*`perfil de segurança para luzes inteligentes`*Comportamentos do Security Profile alterando o`confidenceLevel`de alguns comportamentos e não suprime notificações para todos os comportamentos.

```
aws iot update-security-profile \
    --security-profile-name security-profile-for-smart-lights \
    --behaviors \
     '[{
      "name": "num-messages-sent-ml-behavior",
      "metric": "aws:num-messages-sent",
      "criteria": {
          "mlDetectionConfig": {
              "confidenceLevel" : "HIGH"
          }
      },
      "suppressAlerts": false
  },
  {
      "name": "num-authorization-failures-ml-behavior",
      "metric": "aws:num-authorization-failures",
      "criteria": {
          "mlDetectionConfig": {
              "confidenceLevel" : "HIGH"
          }
      },
      "suppressAlerts": false
  },
  {
```

```
        "name": "num-connection-attempts-ml-behavior",
        "metric": "aws:num-connection-attempts",
        "criteria": {
            "mlDetectionConfig": {
                "confidenceLevel" : "HIGH"
            }
        },
        "suppressAlerts": false
    },
    {
        "name": "num-disconnects-ml-behavior",
        "metric": "aws:num-disconnects",
        "criteria": {
            "mlDetectionConfig": {
                "confidenceLevel" : "LOW"
            }
        },
        "suppressAlerts": false

    }]'
```

Resultado:

```
 {
    "securityProfileName": "security-profile-for-smart-lights",
    "securityProfileArn": "arn:aws:iot:eu-west-1:123456789012:securityprofile/security-
profile-for-smart-lights",
    "behaviors": [
        {
            "name": "num-messages-sent-ml-behavior",
            "metric": "aws:num-messages-sent",
            "criteria": {
                "mlDetectionConfig": {
                    "confidenceLevel": "HIGH"
                }
            }
        },
        {
            "name": "num-authorization-failures-ml-behavior",
            "metric": "aws:num-authorization-failures",
            "criteria": {
                "mlDetectionConfig": {
                    "confidenceLevel": "HIGH"
                }
            }
        },
        {
            "name": "num-connection-attempts-ml-behavior",
            "metric": "aws:num-connection-attempts",
            "criteria": {
                "mlDetectionConfig": {
                    "confidenceLevel": "HIGH"
                }
            },
            "suppressAlerts": false
        },
        {
            "name": "num-disconnects-ml-behavior",
            "metric": "aws:num-disconnects",
            "criteria": {
                "mlDetectionConfig": {
                    "confidenceLevel": "LOW"
                }
            },
```

```
            "suppressAlerts": true
        }
    ],
    "version": 2,
    "creationDate": 1600799559.249,
    "lastModifiedDate": 1600800516.856
}
```

Marque o estado de verificação do alarme

Você pode marcar seus alarmes com estados de verificação para ajudar a classificar alarmes e investigar anomalias.

- Marque seus alarmes com um estado de verificação e uma descrição desse estado. Por exemplo, para definir o estado de verificação de um alarme como Falso positivo, use o seguinte comando:

```
aws iot put-verification-state-on-violation --violation-id 12345  --verification-
state FALSE_POSITIVE --verification-state-description "This is dummy description"  --
endpoint https://us-east-1.iot.amazonaws.com --region us-east-1
```

Resultado:

Nenhum.

Mitigar problemas de dispositivos identificados

1. Usar a`create-thing-group`Para criar um grupo de coisas para a ação de mitigação. No exemplo a seguir, criamos um grupo de coisas chamadoThingGroupForDetectMitigationAction.

```
aws iot create-thing-group —thing-group-name ThingGroupForDetectMitigationAction
```

Resultado:

```
{
 "thingGroupName": "ThingGroupForDetectMitigationAction",
 "thingGroupArn": "arn:aws:iot:us-
east-1:123456789012:thinggroup/ThingGroupForDetectMitigationAction",
 "thingGroupId": "4139cd61-10fa-4c40-b867-0fc6209dca4d"
}
```

2. Em seguida, use o`create-mitigation-action`para criar uma ação de mitigação. No exemplo a seguir, criamos uma ação de mitigação chamadadetect_mitigation_actionCom o ARN da função do IAM usada para aplicar a ação de mitigação. Também definimos o tipo de ação e os parâmetros dessa ação. Nesse caso, nossa mitigação moverá as coisas para o nosso grupo de coisas criado anteriormente chamadoThingGroupForDetectMitigationAction.

```
aws iot create-mitigation-action --action-name detect_mitigation_action \
--role-arn arn:aws:iam::123456789012:role/MitigationActionValidRole \
--action-params \
'{
     "addThingsToThingGroupParams": {
         "thingGroupNames": ["ThingGroupForDetectMitigationAction"],
         "overrideDynamicGroups": false
     }
 }'
```

Resultado:

```
{
 "actionArn": "arn:aws:iot:us-
east-1:123456789012:mitigationaction/detect_mitigation_action",
 "actionId": "5939e3a0-bf4c-44bb-a547-1ab59ffe67c3"
}
```

3. Usar a`start-detect-mitigation-actions-task`Comando para iniciar a tarefa de ações de mitigação.`task-id`,`target`e`actions`São parâmetros obrigatórios.

```
aws iot start-detect-mitigation-actions-task \
    --task-id taskIdForMitigationAction \
    --target '{ "violationIds" : [ "violationId-1", "violationId-2" ] }' \
    --actions "detect_mitigation_action" \
    --include-only-active-violations \
    --include-suppressed-alerts
```

Resultado:

```
{
    "taskId": "taskIdForMitigationAction"
}
```

4. (Opcional) Para exibir execuções de ação de mitigação incluídas em uma tarefa, use o`list-detect-mitigation-actions-executions`comando.

```
aws iot list-detect-mitigation-actions-executions \
    --task-id taskIdForMitigationAction \
    --max-items 5 \
    --page-size 4
```

Resultado:

```
{
    "actionsExecutions": [
        {
            "taskId": "e56ee95e - f4e7 - 459 c - b60a - 2701784290 af",
            "violationId": "214_fe0d92d21ee8112a6cf1724049d80",
            "actionName": "underTest_MAThingGroup71232127",
            "thingName": "cancelDetectMitigationActionsTaskd143821b",
            "executionStartDate": "Thu Jan 07 18: 35: 21 UTC 2021",
            "executionEndDate": "Thu Jan 07 18: 35: 21 UTC 2021",
            "status": "SUCCESSFUL",
        }
    ]
}
```

5. (Opcional) Use o`describe-detect-mitigation-actions-task`Para obter informações sobre uma tarefa de ação de mitigação.

```
aws iot describe-detect-mitigation-actions-task \
    --task-id taskIdForMitigationAction
```

Resultado:

```
{
    "taskSummary": {
```

```
        "taskId": "taskIdForMitigationAction",
        "taskStatus": "SUCCESSFUL",
        "taskStartTime": 1609988361.224,
        "taskEndTime": 1609988362.281,
        "target": {
            "securityProfileName": "security-profile-for-smart-lights",
            "behaviorName": "num-messages-sent-ml-behavior"
        },
        "violationEventOccurrenceRange": {
            "startTime": 1609986633.0,
            "endTime": 1609987833.0
        },
        "onlyActiveViolationsIncluded": true,
        "suppressedAlertsIncluded": true,
        "actionsDefinition": [
            {
                "name": "detect_mitigation_action",
                "id": "5939e3a0-bf4c-44bb-a547-1ab59ffe67c3",
                "roleArn": "arn:aws:iam::123456789012:role/MitigatioActionValidRole",
                "actionParams": {
                    "addThingsToThingGroupParams": {
                        "thingGroupNames": [
                            "ThingGroupForDetectMitigationAction"
                        ],
                        "overrideDynamicGroups": false
                    }
                }
            }
        ],
        "taskStatistics": {
            "actionsExecuted": 0,
            "actionsSkipped": 0,
            "actionsFailed": 0
        }
    }
}
```

6. (Opcional) Para obter uma lista de suas tarefas de ações de mitigação, use o`list-detect-mitigation-actions-tasks`comando.

```
aws iot list-detect-mitigation-actions-tasks \
    --start-time 1609985315 \
    --end-time 1609988915 \
    --max-items 5 \
    --page-size 4
```

Resultado:

```
{
    "tasks": [
        {
            "taskId": "taskIdForMitigationAction",
            "taskStatus": "SUCCESSFUL",
            "taskStartTime": 1609988361.224,
            "taskEndTime": 1609988362.281,
            "target": {
                "securityProfileName": "security-profile-for-smart-lights",
                "behaviorName": "num-messages-sent-ml-behavior"
            },
            "violationEventOccurrenceRange": {
                "startTime": 1609986633.0,
                "endTime": 1609987833.0
            },
```

```
            "onlyActiveViolationsIncluded": true,
            "suppressedAlertsIncluded": true,
            "actionsDefinition": [
                {
                    "name": "detect_mitigation_action",
                    "id": "5939e3a0-bf4c-44bb-a547-1ab59ffe67c3",
                    "roleArn": "arn:aws:iam::123456789012:role/
MitigatioActionValidRole",
                    "actionParams": {
                        "addThingsToThingGroupParams": {
                            "thingGroupNames": [
                                "ThingGroupForDetectMitigationAction"
                            ],
                            "overrideDynamicGroups": false
                        }
                    }
                }
            ],
            "taskStatistics": {
                "actionsExecuted": 0,
                "actionsSkipped": 0,
                "actionsFailed": 0
            }
        }
    ]
}
```

7. (Opcional) Para cancelar uma tarefa de ações de mitigação, use o`cancel-detect-mitigation-actions-task`comando.

```
aws iot cancel-detect-mitigation-actions-task \
    --task-id taskIdForMitigationAction
```

Resultado:

Nenhum.

Personalize quando e como você visualizaAWS IoT Device DefenderAuditoria de resultados

AWS IoT Device Defenderauditoria fornece verificações de segurança periódicas para confirmarAWS IoTdispositivos e recursos estão seguindo as melhores práticas. Para cada verificação, os resultados da auditoria são categorizados como compatíveis ou não compatíveis, em que a não conformidade resulta em ícones de aviso do console. Para reduzir o ruído da repetição de problemas conhecidos, o recurso de supressão de busca de auditoria permite silenciar temporariamente essas notificações de não conformidade.

Você pode suprimir verificações de auditoria selecionadas para um recurso ou conta específico por um período de tempo predeterminado. Um resultado de verificação de auditoria que foi suprimido é categorizado como uma descoberta suprimida, separada das categorias compatíveis e não compatíveis. Essa nova categoria não aciona um alarme como um resultado não compatível. Isso permite reduzir os distúrbios da notificação de não conformidade durante períodos de manutenção conhecidos ou até que uma atualização seja agendada para ser concluída.

Conceitos básicos

As seções a seguir detalham como você pode usar supressões de localização de auditoria para suprimir um`Device certificate expiring`Verifique o console e a CLI. Se você quiser seguir qualquer uma das demonstrações, primeiro você deve criar dois certificados expirantes para o Device Defender detectar.

Use o seguinte para criar seus certificados.

- Criar e registrar um certificado CA (p. 301)
- Criar um certificado de cliente usando o certificado CA. Na etapa 3, defina `odays`parâmetro para`1`.

Se você estiver usando a CLI para criar seus certificados, insira o comando a seguir.

```
openssl x509 -req \
    -in device_cert_csr_filename \
    -CA root_ca_pem_filename \
    -CAkey root_ca_key_filename \
    -CAcreateserial \
    -out device_cert_pem_filename \
    -days 1 -sha256
```

Personalize suas descobertas de auditoria no console

O passo a seguir usa uma conta com dois certificados de dispositivo expirados que acionam uma verificação de auditoria não compatível. Nesse cenário, queremos desativar o aviso porque nossos desenvolvedores estão testando um novo recurso que resolverá o problema. Criamos uma supressão de descoberta de auditoria para cada certificado para impedir que o resultado da auditoria não esteja em conformidade na próxima semana.

1. Primeiro, executaremos uma auditoria sob demanda para mostrar que a verificação do certificado do dispositivo expirado não está em conformidade.

   DoAWS IoTconsole, escolhaDefenderda barra lateral esquerda e depoisAuditoria, e depoisResultados. NoResultados da auditoriaEscolha, escolhaCriar. OCriar uma nova auditoriaA janela é aberta. Escolha Create (Criar).

A partir dos resultados da auditoria sob demanda, podemos ver que “Certificado de dispositivo expirando” não está em conformidade com dois recursos.

2. Agora, gostaríamos de desativar o aviso de verificação não compatível “Certificado de dispositivo expirando” porque nossos desenvolvedores estão testando novos recursos que corrigirão o aviso.

   Da barra lateral esquerda abaixoDefender, escolhaAuditoriaEscolha e, depois, escolhaComo encontrar supressões. NoSupressões de descoberta de auditoriaEscolha, escolhaCriar.

3. NoCriar uma supressão de descoberta de auditoriajanela, precisamos preencher o seguinte.
   - Verificação de auditoria: Selecionamos`Device certificate expiring`, porque essa é a verificação de auditoria que gostaríamos de suprimir.
   - Identificador do recurso: Nós inserimos o ID do certificado do dispositivo de um dos certificados para os quais gostaríamos de suprimir as descobertas de auditoria.
   - Duração da supressão: Selecionamos`1 week`, porque é por quanto tempo gostaríamos de suprimir o`Device certificate expiring`verificação de auditoria para.
   - Description (opcional): Adicionamos uma nota que descreve por que estamos suprimindo essa descoberta de auditoria.

Create an audit finding suppression

Suppressing an audit finding on a specified resource
to the resource for the specified audit check will no l
compliant.

Audit check

Device certificate expiring

Resource identifier

Device certificate id

b4490bd64c5cf85182f3182f1c03e70017e483f17b

Suppression duration

1 week

Description (optional)

Developer updates

Depois de preencher os campos, escolhaCriar. Vemos um banner de sucesso depois que a supressão de descoberta de auditoria foi criada.

4. Suprimimos uma descoberta de auditoria para um dos certificados e agora precisamos suprimir a descoberta de auditoria para o segundo certificado. Poderíamos usar o mesmo método de supressão que usamos na etapa 3, mas usaremos um método diferente para fins de demonstração.

   Da barra lateral esquerda abaixoDefender, escolhaAuditoriaEscolha e, depois, escolhaResultados. NoResultados da auditoria, escolha a auditoria com o recurso incompatível. Em seguida, selecione o recurso emVerificações incompatíveis. No nosso caso, selecionamos “Certificado de dispositivo expirando”.

5. NoCertificado de dispositivo expirandopágina, emPolítica incompatível:Escolha o botão de opção ao lado da descoberta que precisa ser suprimida. Em seguida, escolha aAçõesmenu suspenso e, em seguida, escolha a duração pela qual você gostaria de encontrar para ser suprimida. No nosso caso, escolhemos`1 week`como fizemos para o outro certificado. NoConfirmar supressão, escolhaHabilitar supressão.

2 of 195 device certificates non-compli

Mitigation

Consult your security best practices for how to proc
1. Provision a new certificate and attach it to the de
2. Verify that the new certificate is valid and the dev
3. Mark the old certificate as "INACTIVE" in the AWS
4. Detach the old certificate from the device. (See D

Non-compliant certificate (2)

	Finding
◉	28022a890964e991852c79a28a83eb89
○	dc9b109c705ed7e68588bc54eef86f1c

Vemos um banner de sucesso depois que a supressão de descoberta de auditoria foi criada. Agora, ambas as descobertas de auditoria foram suprimidas por 1 semana, enquanto nossos desenvolvedores trabalham em uma solução para resolver o aviso.

Personalize suas descobertas de auditoria na CLI

O passo a seguir usa uma conta com um certificado de dispositivo expirado que aciona uma verificação de auditoria não compatível. Nesse cenário, queremos desativar o aviso porque nossos desenvolvedores estão testando um novo recurso que resolverá o problema. Criamos uma supressão de descoberta de auditoria para que o certificado impeça que o resultado da auditoria não esteja em conformidade na próxima semana.

Usamos os seguintes comandos da CLI.

- create-audit-supressão
- descrever-supressão de auditoria
- supressão de auditoria de atualização
- supressão de exclusão de auditoria
- supressões de auditoria de lista

1. Use o seguinte comando para habilitar a auditoria.

```
aws iot update-account-audit-configuration \
   --audit-check-configurations "{\"DEVICE_CERTIFICATE_EXPIRING_CHECK\":{\"enabled
\":true}}"
```

   Resultado:

   Nenhum.

2. Use o seguinte comando para executar uma auditoria sob demanda que seja direcionada ao`DEVICE_CERTIFICATE_EXPIRING_CHECK`Verificação de auditoria.

```
aws iot start-on-demand-audit-task \
    --target-check-names DEVICE_CERTIFICATE_EXPIRING_CHECK
```

   Resultado:

```
{
    "taskId": "787ed873b69cb4d6cdbae6ddd06996c5"
}
```

3. Usar aDescribe-account-audit-configurationpara descrever a configuração de auditoria. Queremos confirmar que ativamos a verificação de auditoria para`DEVICE_CERTIFICATE_EXPIRING_CHECK`.

```
aws iot describe-account-audit-configuration
```

   Resultado:

```
{
    "roleArn": "arn:aws:iam::<accountid>:role/service-role/project",
    "auditNotificationTargetConfigurations": {
        "SNS": {
```

```
            "targetArn": "arn:aws:sns:us-east-1:<accountid>:project_sns",
            "roleArn": "arn:aws:iam::<accountid>:role/service-role/project",
            "enabled": true
        }
    },
    "auditCheckConfigurations": {
        "AUTHENTICATED_COGNITO_ROLE_OVERLY_PERMISSIVE_CHECK": {
            "enabled": false
        },
        "CA_CERTIFICATE_EXPIRING_CHECK": {
            "enabled": false
        },
        "CA_CERTIFICATE_KEY_QUALITY_CHECK": {
            "enabled": false
        },
        "CONFLICTING_CLIENT_IDS_CHECK": {
            "enabled": false
        },
        "DEVICE_CERTIFICATE_EXPIRING_CHECK": {
            "enabled": true
        },
        "DEVICE_CERTIFICATE_KEY_QUALITY_CHECK": {
            "enabled": false
        },
        "DEVICE_CERTIFICATE_SHARED_CHECK": {
            "enabled": false
        },
        "IOT_POLICY_OVERLY_PERMISSIVE_CHECK": {
            "enabled": true
        },
        "IOT_ROLE_ALIAS_ALLOWS_ACCESS_TO_UNUSED_SERVICES_CHECK": {
            "enabled": false
        },
        "IOT_ROLE_ALIAS_OVERLY_PERMISSIVE_CHECK": {
            "enabled": false
        },
        "LOGGING_DISABLED_CHECK": {
            "enabled": false
        },
        "REVOKED_CA_CERTIFICATE_STILL_ACTIVE_CHECK": {
            "enabled": false
        },
        "REVOKED_DEVICE_CERTIFICATE_STILL_ACTIVE_CHECK": {
            "enabled": false
        },
        "UNAUTHENTICATED_COGNITO_ROLE_OVERLY_PERMISSIVE_CHECK": {
            "enabled": false
        }
    }
}
```

`DEVICE_CERTIFICATE_EXPIRING_CHECK`deve ter um valor de`true`.

4. Usar a`list-audit-task`para identificar as tarefas de auditoria concluídas.

```
aws iot list-audit-tasks \
    --task-status "COMPLETED" \
    --start-time 2020-07-31 \
    --end-time 2020-08-01
```

Resultado:

```
{
```

```
    "tasks": [
        {
            "taskId": "787ed873b69cb4d6cdbae6ddd06996c5",
            "taskStatus": "COMPLETED",
            "taskType": "SCHEDULED_AUDIT_TASK"
        }
    ]
}
```

O`taskId`da auditoria que você executou na etapa 1 deve ter um`taskStatus`do`COMPLETED`.

5. Usar adescribe-audit-taskPara obter detalhes sobre a auditoria concluída usando a`taskId`Saída da etapa anterior. Este comando lista detalhes sobre sua auditoria.

```
aws iot describe-audit-task \
    --task-id "787ed873b69cb4d6cdbae6ddd06996c5"
```

Resultado:

```
{
    "taskStatus": "COMPLETED",
    "taskType": "SCHEDULED_AUDIT_TASK",
    "taskStartTime": 1596168096.157,
    "taskStatistics": {
        "totalChecks": 1,
        "inProgressChecks": 0,
        "waitingForDataCollectionChecks": 0,
        "compliantChecks": 0,
        "nonCompliantChecks": 1,
        "failedChecks": 0,
        "canceledChecks": 0
    },
    "scheduledAuditName": "AWSIoTDeviceDefenderDailyAudit",
    "auditDetails": {
        "DEVICE_CERTIFICATE_EXPIRING_CHECK": {
            "checkRunStatus": "COMPLETED_NON_COMPLIANT",
            "checkCompliant": false,
            "totalResourcesCount": 195,
            "nonCompliantResourcesCount": 2
        }
    }
}
```

6. Usar alist-audit-descobertaspara localizar o ID do certificado não compatível para que possamos suspender os alertas de auditoria desse recurso.

```
aws iot list-audit-findings \
    --start-time 2020-07-31 \
    --end-time 2020-08-01
```

Resultado:

```
{
    "findings": [
        {
            "findingId": "296ccd39f806bf9d8f8de20d0ceb33a1",
            "taskId": "787ed873b69cb4d6cdbae6ddd06996c5",
            "checkName": "DEVICE_CERTIFICATE_EXPIRING_CHECK",
            "taskStartTime": 1596168096.157,
            "findingTime": 1596168096.651,
            "severity": "MEDIUM",
```

```
            "nonCompliantResource": {
                "resourceType": "DEVICE_CERTIFICATE",
                "resourceIdentifier": {
                    "deviceCertificateId": "b4490<shortened>"
                },
                "additionalInfo": {
                "EXPIRATION_TIME": "1582862626000"
                }
            },
            "reasonForNonCompliance": "Certificate is past its expiration.",
            "reasonForNonComplianceCode": "CERTIFICATE_PAST_EXPIRATION",
            "isSuppressed": false
        },
        {
            "findingId": "37ecb79b7afb53deb328ec78e647631c",
            "taskId": "787ed873b69cb4d6cdbae6ddd06996c5",
            "checkName": "DEVICE_CERTIFICATE_EXPIRING_CHECK",
            "taskStartTime": 1596168096.157,
            "findingTime": 1596168096.651,
            "severity": "MEDIUM",
            "nonCompliantResource": {
                "resourceType": "DEVICE_CERTIFICATE",
                "resourceIdentifier": {
                    "deviceCertificateId": "c7691<shortened>"
                },
                "additionalInfo": {
                "EXPIRATION_TIME": "1583424717000"
                }
            },
            "reasonForNonCompliance": "Certificate is past its expiration.",
            "reasonForNonComplianceCode": "CERTIFICATE_PAST_EXPIRATION",
            "isSuppressed": false
        }
    ]
}
```

7. Usar acreate-audit-supressãocomando para suprimir notificações para o`DEVICE_CERTIFICATE_EXPIRING_CHECK`verificação de auditoria de um certificado de dispositivo com o id*c7691e<shortened>*até*20/08/2020*.

```
aws iot create-audit-suppression \
    --check-name DEVICE_CERTIFICATE_EXPIRING_CHECK \
    --resource-identifier deviceCertificateId="c7691e<shortened>" \
    --no-suppress-indefinitely \
    --expiration-date 2020-08-20
```

8. Usar asupressão de auditoria de listapara confirmar a configuração de supressão de auditoria e obter detalhes sobre a supressão.

```
aws iot list-audit-suppressions
```

Resultado:

```
{
    "suppressions": [
        {
        "checkName": "DEVICE_CERTIFICATE_EXPIRING_CHECK",
            "resourceIdentifier": {
                "deviceCertificateId": "c7691e<shortened>"
            },
        "expirationDate": 1597881600.0,
        "suppressIndefinitely": false
        }
```

```
    ]
}
```

9. Osupressão de auditoria de atualizaçãopode ser usado para atualizar a supressão de busca de auditoria. O exemplo abaixo atualiza a`expiration-date`para`08/21/20`.

```
aws iot update-audit-suppression \
    --check-name DEVICE_CERTIFICATE_EXPIRING_CHECK \
    --resource-identifier deviceCertificateId=c7691e<shortened> \
    --no-suppress-indefinitely \
    --expiration-date 2020-08-21
```

10. Osupressão de exclusão de auditoriapode ser usado para remover uma supressão de descoberta de auditoria.

```
aws iot delete-audit-suppression \
    --check-name DEVICE_CERTIFICATE_EXPIRING_CHECK \
    --resource-identifier deviceCertificateId="c7691e<shortened>"
```

Para confirmar a exclusão, use asupressões de auditoria de listacomando.

```
aws iot list-audit-suppressions
```

Resultado:

```
{
 "suppressions": []
}
```

Neste tutorial, mostramos como suprimir um`Device certificate expiring`Verifique o console e a CLI. Para obter mais informações sobre supressão de descoberta de auditoria, consulteSupressões de descoberta de auditoria (p. 984)

Auditoria

A auditoria do AWS IoT Device Defender analisa configurações e políticas relacionadas a contas e dispositivos para garantir que as medidas de segurança estejam em vigor. Uma auditoria pode ajudar você a detectar todos os desvios das práticas recomendadas de segurança ou das políticas de acesso (por exemplo, vários dispositivos usando a mesma identidade ou políticas excessivamente permissivas que permitem que um dispositivo leia e atualize dados para muitos outros dispositivos). Você pode executar auditorias conforme necessário (auditorias sob demanda) ou programá-las para execução periódica (auditorias programadas).

Uma auditoria do AWS IoT Device Defender realiza um conjunto de verificações predefinidas para melhores práticas de segurança e vulnerabilidades de dispositivos de IoT. Os exemplos de verificações predefinidas incluem políticas que concedem permissões para leitura e atualização de dados em vários dispositivos, dispositivos que compartilham uma identidade (certificado X.509) ou certificados que estão prestes a expirar ou foram revogados, mas ainda estão ativos.

Gravidade do problema

A gravidade do problema indica o nível de preocupação associado a cada instância identificada de não conformidade e o tempo recomendado para a correção.

Critical

As verificações de auditoria não compatíveis com essa gravidade identificam problemas que exigem atenção urgente. Problemas críticos geralmente permitem que usuários maliciosos com pouca sofisticação e sem conhecimento privilegiado ou credenciais especiais ganhem facilmente acesso ou controle de seus ativos.

Alto

As verificações de auditoria não compatíveis com essa gravidade exigem investigação urgente e planejamento após problemas críticos serem resolvidos. Assim como problemas críticos, os problemas de alta gravidade geralmente fornecem aos usuários maliciosos acesso ou controle de seus ativos. No entanto, problemas de alta gravidade são geralmente mais difíceis de explorar. Eles podem exigir ferramentas especiais, conhecimento privilegiado ou configurações específicas.

Médio

As verificações de auditoria não compatíveis com essa gravidade apresentam problemas que precisam de atenção como parte da manutenção contínua da postura de segurança. Problemas de gravidade média podem causar impacto operacional negativo, como interrupções não planejadas devido ao mau funcionamento dos controles de segurança. Esses problemas também podem fornecer aos usuários maliciosos acesso ou controle limitado de seus ativos, ou podem facilitar partes das ações mal-intencionadas deles.

Baixo

Verificações de auditoria não compatíveis com essa gravidade geralmente indicam que as práticas recomendadas de segurança foram ignoradas ou negligenciadas. Embora talvez não causem um impacto imediato na segurança por conta própria, esses lapsos podem ser explorados por usuários maliciosos. Assim como problemas de gravidade média, os problemas de baixa gravidade exigem atenção como parte da manutenção contínua da postura de segurança.

Próximas etapas

Para entender os tipos de verificações de auditoria que podem ser executadas, consulte Verificações da auditoria (p. 931). Para obter informações sobre cotas de serviço que se aplicam a auditorias, consulte Cotas de serviço.

Verificações da auditoria

Note

Quando você habilita uma verificação, a coleta dos dados é iniciada imediatamente. Se houver um grande volume de dados na conta para serem coletados, os resultados da verificação poderão não ficar disponíveis por algum tempo depois de habilitada.

Há suporte para as seguintes verificações de auditoria:

- Certificado de CA revogado, mas os certificados de dispositivo ainda ativos (p. 932)
- Certificado de dispositivo compartilhado (p. 932)
- Qualidade da chave do certificado do dispositivo (p. 933)
- Qualidade da chave de certificado CA (p. 935)
- Papel Cognito não autenticado excessivamente permissivo (p. 936)
- Função autenticada do Cognito excessivamente permissiva (p. 941)
- AWS IoTPolíticas excessivamente permissivas (p. 948)
- Alias de função excessivamente permissivo (p. 952)

- O alias de função permite acesso a serviços não utilizados (p. 953)
- Certificado CA expirar (p. 954)
- IDs do cliente MQTT conflitantes (p. 954)
- Certificado de dispositivo expirar (p. 955)
- Certificado de dispositivo revogado ainda ativo (p. 956)
- Como desabilitar o registro (p. 957)

Certificado de CA revogado, mas os certificados de dispositivo ainda ativos

Um certificado da CA foi revogado, mas ainda está ativo na AWS IoT.

Essa verificação aparece como`REVOKED_CA_CERTIFICATE_STILL_ACTIVE_CHECK`na CLI e a API.

Gravidade: Critical

Detalhes

Um certificado da CA está marcado como revogado na lista de revogação de certificados mantida pela autoridade emissora, mas ainda está marcado como ACTIVE ou PENDING_TRANSFER na AWS IoT.

Os códigos de motivo a seguir são retornados quando essa verificação encontra um certificado da CA não compatível:

- CERTIFICATE_REVOKED_BY_ISSUER

Por que isso importa?

Um certificado da CA não deve mais ser usado para assinar certificados de dispositivo. Ele pode ter sido revogado porque foi comprometido. Dispositivos recém-adicionados com certificados assinados usando esse certificado da CA podem representar uma ameaça à segurança.

Como corrigir

1. Use UpdateCACertificate para marcar o certificado da CA como INACTIVE no AWS IoT. Você também pode usar ações de mitigação para:
   - Aplicar a ação de mitigação `UPDATE_CA_CERTIFICATE` em suas descobertas de auditoria para fazer essa mudança.
   - Aplicar `oPUBLISH_FINDINGS_TO_SNS`Ação de mitigação para implementar uma resposta personalizada em resposta à mensagem do Amazon SNS.

   Para obter mais informações, consulte Ações de mitigação (p. 1045).
2. Revise as atividades de registro de certificado de dispositivo pelo período após o qual o certificado da CA foi revogado e considere a possibilidade de revogar todos os certificados de dispositivos que podem ter sido emitidos com ele durante esse período. Você pode usar ListCertificatesByCA para listar os certificados de dispositivo assinados pelo certificado da CA e UpdateCertificate para revogar um certificado de dispositivo.

Certificado de dispositivo compartilhado

Várias conexões simultâneas usando o mesmo certificado X.509 para ser autenticadas com a AWS IoT.

Essa verificação aparece como`DEVICE_CERTIFICATE_SHARED_CHECK`na CLI e a API.

Gravidade: Critical

Detalhes

Quando executada como parte de uma auditoria sob demanda, essa verificação examina os certificados e os IDs de clientes que foram usados pelos dispositivos para se conectar durante os 31 dias anteriores ao início da auditoria até 2 horas anteriores ao início da auditoria até 2 horas anteriores a execução da verificação. Para auditorias programadas, essa verificação examina dados de 2 horas anteriores à última vez em que a auditoria foi executada até 2 horas anteriores à hora em que essa instância da auditoria foi iniciada. Se você tiver tomado medidas para atenuar essa condição durante a verificação, observe quando as conexões simultâneas foram estabelecidas para determinar se o problema persiste.

Os códigos de motivo a seguir são retornados quando essa verificação encontra um certificado não compatível:

- CERTIFICATE_SHARED_BY_MULTIPLE_DEVICES

Além disso, as descobertas retornadas por essa verificação incluem o ID do certificado compartilhado, os IDs dos clientes que usam o certificado para se conectar e os horários de conexão/desconexão. Os resultados mais recentes são listados primeiro.

Por que isso importa?

Cada dispositivo deve ter um certificado exclusivo para se autenticar na AWS IoT. Se vários dispositivos usam o mesmo certificado, isso pode indicar que um dispositivo foi comprometido. A sua identidade pode ter sido clonada para comprometer ainda mais o sistema.

Como corrigir

Verifique se o certificado de dispositivo não foi comprometido. Se foi, siga as práticas recomendadas de segurança para atenuar a situação.

Se estiver usando o mesmo certificado em vários dispositivos, você poderá:

1. Provisionar novos certificados exclusivos e anexá-los a cada dispositivo.
2. Verificar se os novos certificados são válidos e se os dispositivos podem usá-los para se conectar.
3. Use UpdateCertificate para marcar o certificado antigo como REVOKED no AWS IoT. Você também pode usar ações de mitigação para fazer o seguinte:
   - Aplicar a ação de mitigação `UPDATE_DEVICE_CERTIFICATE` em suas descobertas de auditoria para fazer essa mudança.
   - Aplicar a ação de mitigação `ADD_THINGS_TO_THING_GROUP` para adicionar o dispositivo a um grupo, onde é possível executar uma ação sobre ele.
   - Aplicar `OPUBLISH_FINDINGS_TO_SNS`Se você desejar implementar uma resposta personalizada em resposta à mensagem do Amazon SNS.

   Para obter mais informações, consulte Ações de mitigação (p. 1045).
4. Desanexe o antigo certificado de todos os dispositivos.

Qualidade da chave do certificado do dispositivo

A AWS IoT frequentemente confia na autenticação mútua TLS usando certificados X.509 para autenticação no agente de mensagens da AWS IoT. Esses certificados e os respectivos certificados de

autoridade de certificação devem ser registrados na conta do AWS IoT antes de serem utilizados. O AWS IoT executa verificações básicas de sanidade nesses certificados quando eles são registrados. Essas verificações incluem:

- Devem estar em um formato válido.
- Devem ser assinados por uma autoridade de certificação registrada.
- Ainda devem estar dentro do prazo de validade (ou seja, não podem ter expirado).
- Os tamanhos de chave criptográfica devem atender ao tamanho mínimo necessário (para chaves RSA, devem ter 2048 bits ou mais).

Essa verificação de auditoria fornece os seguintes testes adicionais da qualidade da chave criptográfica:

- CVE-2008-0166 — verifique se a chave foi gerada usando OpenSSL 0.9.8c-1 até versões anteriores a 0.9.8g-9 em um sistema operacional baseado em Debian. Essas versões do OpenSSL usam um gerador de números aleatórios que gera números previsíveis, facilitando para invasores remotos realizar ataques de adivinhação de força bruta contra chaves criptográficas.
- CVE-2017-15361 — verifique se a chave foi gerada pela biblioteca Infineon RSA 1.02.013 no firmware do Infineon Trusted Platform Module (TPM), como versões anteriores a 000000000422 — 4.34, anteriores a 00000000062b — 6.43 e anteriores a 00008521 — 133.33. Essa biblioteca manipula incorretamente a geração de chaves RSA, tornando mais fácil para os invasores derrotarem alguns mecanismos de proteção criptográfica por meio de ataques direcionados. Exemplos de tecnologias afetadas incluem BitLocker com TPM 1.2, geração de chaves PGP YubiKey 4 (anterior a 4.3.5) e o recurso de criptografia de dados de usuário em cache no Chrome OS.

O AWS IoT Device Defender relata certificados como não compatíveis quando eles falham nesses testes.

Essa verificação aparece como`DEVICE_CERTIFICATE_KEY_QUALITY_CHECK`na CLI e a API.

Gravidade: Critical

Detalhes

Essa verificação se aplica a certificados de dispositivo que estão ACTIVE ou PENDING_TRANSFER.

Os códigos de motivo a seguir são retornados quando essa verificação encontra um certificado não compatível:

- CERTIFICATE_KEY_VULNERABILITY_CVE-2017-15361
- CERTIFICATE_KEY_VULNERABILITY_CVE-2008-0166

Por que isso importa?

Quando um dispositivo usa um certificado vulnerável, os invasores podem comprometer mais facilmente esse dispositivo.

Como corrigir

Atualize seus certificados de dispositivo para substituir aqueles com vulnerabilidades conhecidas.

Se você estiver usando o mesmo certificado em vários dispositivos, poderá:

1. Provisionar novos certificados exclusivos e anexá-los a cada dispositivo.
2. Verificar se os novos certificados são válidos e se os dispositivos podem usá-los para se conectar.

3. Use UpdateCertificate para marcar o certificado antigo como REVOKED no AWS IoT. Você também pode usar ações de mitigação para:
   - Aplicar a ação de mitigação `UPDATE_DEVICE_CERTIFICATE` em suas descobertas de auditoria para fazer essa mudança.
   - Aplicar a ação de mitigação `ADD_THINGS_TO_THING_GROUP` para adicionar o dispositivo a um grupo, onde é possível executar uma ação sobre ele.
   - Aplicar `oPUBLISH_FINDINGS_TO_SNS`Se você desejar implementar uma resposta personalizada em resposta à mensagem do Amazon SNS.

   Para obter mais informações, consulte Ações de mitigação (p. 1045).
4. Desanexe o antigo certificado de todos os dispositivos.

Qualidade da chave de certificado CA

A AWS IoT frequentemente confia na autenticação mútua TLS usando certificados X.509 para autenticação no agente de mensagens da AWS IoT. Esses certificados e os respectivos certificados da autoridade de certificação devem ser registrados na sua conta do AWS IoT antes de serem utilizados. O AWS IoT executa verificações básicas de sanidade nesses certificados quando eles são registrados, incluindo:

- Os certificados estão em um formato válido.
- Os certificados estão dentro do período de validade (em outras palavras, não expiraram).
- Os tamanhos de chave criptográfica atendem ao tamanho mínimo necessário (para chaves RSA, devem ter 2048 bits ou mais).

Essa verificação de auditoria fornece os seguintes testes adicionais da qualidade da chave criptográfica:

- CVE-2008-0166 — verifique se a chave foi gerada usando OpenSSL 0.9.8c-1 até versões anteriores a 0.9.8g-9 em um sistema operacional baseado em Debian. Essas versões do OpenSSL usam um gerador de números aleatórios que gera números previsíveis, facilitando para invasores remotos realizar ataques de adivinhação de força bruta contra chaves criptográficas.
- CVE-2017-15361 — verifique se a chave foi gerada pela biblioteca Infineon RSA 1.02.013 no firmware do Infineon Trusted Platform Module (TPM), como versões anteriores a 000000000422 — 4.34, anteriores a 00000000062b — 6.43 e anteriores a 00008521 — 133.33. Essa biblioteca manipula incorretamente a geração de chaves RSA, tornando mais fácil para os invasores derrotarem alguns mecanismos de proteção criptográfica por meio de ataques direcionados. Exemplos de tecnologias afetadas incluem BitLocker com TPM 1.2, geração de chaves PGP YubiKey 4 (anterior a 4.3.5) e o recurso de criptografia de dados de usuário em cache no Chrome OS.

O AWS IoT Device Defender relata certificados como não compatíveis quando eles falham nesses testes.

Essa verificação aparece como`CA_CERTIFICATE_KEY_QUALITY_CHECK`na CLI e a API.

Gravidade: Critical

Detalhes

Essa verificação se aplica a certificados da CA que estão ACTIVE ou PENDING_TRANSFER.

Os códigos de motivo a seguir são retornados quando essa verificação encontra um certificado não compatível:

- CERTIFICATE_KEY_VULNERABILITY_CVE-2017-15361

- CERTIFICATE_KEY_VULNERABILITY_CVE-2008-0166

Por que isso importa?

Dispositivos recém-adicionados usando esse certificado CA podem representar uma ameaça à segurança.

Como corrigir

1. Use UpdateCACertificate para marcar o certificado da CA como INACTIVE no AWS IoT. Você também pode usar ações de mitigação para:
   - Aplicar a ação de mitigação `UPDATE_CA_CERTIFICATE` em suas descobertas de auditoria para fazer essa mudança.
   - Aplicar `oPUBLISH_FINDINGS_TO_SNS`Se você desejar implementar uma resposta personalizada em resposta à mensagem do Amazon SNS.

   Para obter mais informações, consulte Ações de mitigação (p. 1045).
2. Revise as atividades de registro de certificado de dispositivo pelo período após o qual o certificado da CA foi revogado e considere a possibilidade de revogar todos os certificados de dispositivos que podem ter sido emitidos com ele durante esse período. (Use ListCertificatesByCA para listar os certificados de dispositivo assinados pelo certificado da CA e UpdateCertificate para revogar um certificado de dispositivo.)

Papel Cognito não autenticado excessivamente permissivo

Uma política anexada a uma função do grupo de identidades não autenticadas do Amazon Cognito é considerada excessivamente permissiva porque ela concede permissão para executar qualquer um dos seguintesAWS IoTações:

- Gerenciar ou modificar coisas.
- Ler dados administrativos das coisas.
- Gerenciar dados ou recursos não relacionados a coisas.

Ou, porque ela concede permissão para executar as seguintes ações da AWS IoT em um amplo conjunto de dispositivos:

- Usar MQTT para se conectar, publicar ou assinar tópicos reservados (incluindo sombra ou dados de execução de trabalhos).
- Usar comandos de API para ler ou modificar shadow ou dados de execução de trabalhos.

Em geral, os dispositivos que se conectam usando uma função do grupo de identidades não autenticadas do Amazon Cognito só devem ter permissão limitada para publicar e assinar tópicos do MQTT específicos de coisas ou usar os comandos de API para ler e modificar dados específicos de coisas relacionados a dados de execução de trabalhos ou shadow.

Essa verificação aparece como`UNAUTHENTICATED_COGNITO_ROLE_OVERLY_PERMISSIVE_CHECK`na CLI e a API.

Gravidade: Critical

Detalhes

Para esta verificação,AWS IoT Device DefenderAudita todos os grupos de identidades do Amazon Cognito do que foram usados para se conectar aoAWS IoTcorretor de mensagens durante os 31 dias antes

da execução da auditoria. Todos os grupos de identidades do Amazon Cognito a partir dos quais uma identidade autenticada do Amazon Cognito conectada são incluídos na auditoria.

Os códigos de motivo a seguir são retornados quando essa verificação encontra uma função de grupo de identidades não autenticadas do Amazon Cognito incompatível:

- ALLOWS_ACCESS_TO_IOT_ADMIN_ACTIONS
- ALLOWS_BROAD_ACCESS_TO_IOT_DATA_PLANE_ACTIONS

Por que isso importa?

Como identidades não autenticadas nunca são autenticadas pelo usuário, elas representam um risco muito maior do que as identidades autenticadas do Amazon Cognito. Se uma identidade não autenticada estiver comprometida, ela poderá usar ações administrativas para modificar as configurações da conta, excluir recursos ou obter acesso a dados confidenciais. Ou, com amplo acesso às configurações de dispositivo, poderá acessar ou modificar shadows e trabalhos de todos os dispositivos em sua conta. Um usuário convidado pode usar as permissões para comprometer toda a sua frota ou ativar um ataque DDOS com mensagens.

Como corrigir

Uma política anexada a uma função do grupo de identidades não autenticadas do Amazon Cognito deve conceder somente as permissões necessárias para um dispositivo fazer seu trabalho. Recomendamos as seguintes etapas:

1. Criar uma nova função compatível.
2. Criar um novo grupo de identidades do Amazon Cognito e anexar a função compatível a ele.
3. Verificar se suas identidades podem acessar a AWS IoT usando o novo grupo.
4. Após a conclusão da verificação, anexe a função compatível ao grupo de identidades do Amazon Cognito que foi sinalizado como incompatível.

Você também pode usar ações de mitigação para:

- Aplicar `oPUBLISH_FINDINGS_TO_SNS`Ação de mitigação para implementar uma resposta personalizada em resposta à mensagem do Amazon SNS.

Para obter mais informações, consulte Ações de mitigação (p. 1045).

Gerenciar ou modificar coisas

As ações de API da AWS IoT a seguir são usadas para gerenciar ou modificar coisas. A permissão para executar essas ações não deve ser concedida a dispositivos que se conectam por meio de um grupo de identidades não autenticadas do Amazon Cognito.

- `AddThingToThingGroup`
- `AttachThingPrincipal`
- `CreateThing`
- `DeleteThing`
- `DetachThingPrincipal`
- `ListThings`
- `ListThingsInThingGroup`
- `RegisterThing`
- `RemoveThingFromThingGroup`

- `UpdateThing`
- `UpdateThingGroupsForThing`

Qualquer função que conceda permissão para executar essas ações em até mesmo um único recurso é considerada não compatível.

Ler dados administrativos das coisas

As seguintes ações da API da AWS IoT são usadas para ler ou modificar dados de coisa. Os dispositivos que se conectam por meio de um grupo de identidades não autenticadas do Amazon Cognito não devem receber permissão para executar essas ações.

- `DescribeThing`
- `ListJobExecutionsForThing`
- `ListThingGroupsForThing`
- `ListThingPrincipals`

Example

- incompatível:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
          "iot:DescribeThing",
          "iot:ListJobExecutionsForThing",
          "iot:ListThingGroupsForThing",
          "iot:ListThingPrincipals"
      ],
      "Resource": [
        "arn:aws:iot:region:account-id:/thing/MyThing"
      ]
    }
  ]
}
```

Permite que o dispositivo execute as ações especificadas mesmo que sejam concedidas para uma coisa apenas.

Gerenciar não coisas

Os dispositivos que se conectam por meio de um grupo de identidades não autenticadas do Amazon Cognito não devem receber permissão para executarAWS IoTAções de API diferentes daquelas discutidas nessas seções. Você pode gerenciar sua conta com um aplicativo que se conecta por meio de um grupo de identidades não autenticadas do Amazon Cognito criando um grupo de identidades separado não usado pelos dispositivos.

Assinar/publicar em tópicos do MQTT

As mensagens do MQTT são enviadas por meio do agente de mensagens do AWS IoT e são usadas pelos dispositivos para executar muitas ações, incluindo acessar e modificar o estado de sombra e o estado de execução de trabalhos. Uma política que concede permissão para um dispositivo se conectar, publicar ou assinar mensagens do MQTT deve restringir essas ações a recursos específicos da seguinte forma:

Conecte-se

- incompatível:

```
arn:aws:iot:region:account-id:client/*
```

O curinga * permite que qualquer dispositivo se conecte ao AWS IoT.

```
arn:aws:iot:region:account-id:client/${iot:ClientId}
```

A menos que `iot:Connection.Thing.IsAttached` seja definido como true nas chaves de condição, é equivalente ao curinga * no exemplo anterior.

- compatível:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [ "iot:Connect" ],
      "Resource": [
        "arn:aws:iot:region:account-id:client/${iot:Connection.Thing.ThingName}"
      ],
      "Condition": {
        "Bool": { "iot:Connection.Thing.IsAttached": "true" }
      }
    }
  ]
}
```

A especificação de recurso contém uma variável que corresponde ao nome do dispositivo usado para se conectar. A declaração de condição restringe ainda mais a permissão, verificando se o certificado usado pelo cliente MQTT corresponde ao que é anexado à coisa com o nome usado.

Publicar

- incompatível:

```
arn:aws:iot:region:account-id:topic/$aws/things/*/shadow/update
```

Isso permite que o dispositivo atualize o shadow de qualquer dispositivo (* = todos os dispositivos).

```
arn:aws:iot:region:account-id:topic/$aws/things/*
```

Isso permite que o dispositivo leia, atualize ou exclua a sombra de qualquer dispositivo.

- compatível:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [ "iot:Publish" ],
      "Resource": [
        "arn:aws:iot:region:account-id:topic/$aws/things/
${iot:Connection.Thing.ThingName}/shadow/*"
      ],
    }
  ]
```

```
}
```

A especificação do recurso contém um caractere curinga, mas apenas corresponde a qualquer tópico relacionado a shadow para o dispositivo cujo nome da coisa é usado para se conectar.

Assinar

- incompatível:

```
arn:aws:iot:region:account-id:topicfilter/$aws/things/*
```

Isso permite que o dispositivo assine tópicos de shadow ou de trabalho reservados para todos os dispositivos.

```
arn:aws:iot:region:account-id:topicfilter/$aws/things/*
```

O mesmo do exemplo anterior, mas usando o curinga #.

```
arn:aws:iot:region:account-id:topicfilter/$aws/things/+/shadow/update
```

Isso permite que o dispositivo veja as atualizações de shadow de qualquer dispositivo (+ = todos os dispositivos).

- compatível:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [ "iot:Subscribe" ],
      "Resource": [
        "arn:aws:iot:region:account-id:topicfilter/$aws/things/
${iot:Connection.Thing.ThingName}/shadow/*"
        "arn:aws:iot:region:account-id:topicfilter/$aws/things/
${iot:Connection.Thing.ThingName}/jobs/*"
      ],
    }
  ]
}
```

As especificações de recursos contêm caracteres curinga, mas eles apenas correspondem a qualquer tópico relacionado a shadow e a trabalho para o dispositivo cujo nome da coisa é usado para se conectar.

Receber

- compatível:

```
arn:aws:iot:region:account-id:topicfilter/$aws/things/*
```

Isso é permitido porque o dispositivo só pode receber mensagens de tópicos nos quais ele tem permissão para assinar.

Ler/modificar dados de sombra ou de trabalho

Uma política que concede permissão para um dispositivo executar uma ação de API para acessar ou modificar shadows de dispositivos ou dados de execução de trabalhos deve restringir essas ações a recursos específicos. Estas são as ações da API:

- `DeleteThingShadow`
- `GetThingShadow`
- `UpdateThingShadow`
- `DescribeJobExecution`
- `GetPendingJobExecutions`
- `StartNextPendingJobExecution`
- `UpdateJobExecution`

Example

- incompatível:

```
arn:aws:iot:region:account-id:thing/*
```

  Isso permite que o dispositivo realize a ação especificada em qualquer coisa.

- compatível:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
          "iot:DeleteThingShadow",
          "iot:GetThingShadow",
          "iot:UpdateThingShadow",
          "iot:DescribeJobExecution",
          "iot:GetPendingJobExecutions",
          "iot:StartNextPendingJobExecution",
          "iot:UpdateJobExecution"
      ],
      "Resource": [
        "arn:aws:iot:region:account-id:/thing/MyThing1",
        "arn:aws:iot:region:account-id:/thing/MyThing2"
      ]
    }
  ]
}
```

  Isso permite que o dispositivo execute ações específicas em duas coisas apenas.

Função autenticada do Cognito excessivamente permissiva

Uma política anexada a uma função do grupo de identidades autenticadas do Amazon Cognito é considerada excessivamente permissiva porque ela concede permissão para executar o seguinteAWS IoTações:

- Gerenciar ou modificar coisas.
- Gerenciar dados ou recursos não relacionados a coisas.

Ou, porque ela concede permissão para executar as seguintes ações da AWS IoT em um amplo conjunto de dispositivos:

- Ler dados administrativos das coisas.

- Usar MQTT para se conectar/publicar/assinar tópicos reservados (incluindo shadow ou dados de execução de trabalhos).
- Usar comandos de API para ler ou modificar shadow ou dados de execução de trabalhos.

Em geral, os dispositivos que se conectam usando uma função do grupo de identidades autenticadas do Amazon Cognito só devem ter permissão limitada para ler dados administrativos específicos de coisas, publicar e assinar tópicos do MQTT específicos de coisas ou usar os comandos de API para ler e modificar dados específicos de coisas relacionados a shadow ou trabalhar Dados de execução.

Essa verificação aparece como`AUTHENTICATED_COGNITO_ROLE_OVERLY_PERMISSIVE_CHECK`na CLI e a API.

Gravidade: Critical

Detalhes

Para esta verificação,AWS IoT Device DefenderAudita todos os grupos de identidades do Amazon Cognito do que foram usados para se conectar aoAWS IoTcorretor de mensagens durante os 31 dias antes da execução da auditoria. Todos os grupos de identidades do Amazon Cognito a partir dos quais uma identidade autenticada do Amazon Cognito conectada são incluídos na auditoria.

Os códigos de motivo a seguir são retornados quando essa verificação encontra uma função de grupo de identidades autenticadas do Amazon Cognito incompatível:

- ALLOWS_BROAD_ACCESS_TO_IOT_THING_ADMIN_READ_ACTIONS
- ALLOWS_ACCESS_TO_IOT_NON_THING_ADMIN_ACTIONS
- ALLOWS_ACCESS_TO_IOT_THING_ADMIN_WRITE_ACTIONS

Por que isso importa?

Se uma identidade autenticada estiver comprometida, ela poderá usar ações administrativas para modificar as configurações da conta, excluir recursos ou obter acesso a dados confidenciais.

Como corrigir

Uma política anexada a uma função do grupo de identidades autenticadas do Amazon Cognito deve conceder somente as permissões necessárias para um dispositivo fazer seu trabalho. Recomendamos as seguintes etapas:

1. Criar uma nova função compatível.
2. Criar um novo grupo de identidades do Amazon Cognito e anexar a função compatível a ele.
3. Verificar se suas identidades podem acessar a AWS IoT usando o novo grupo.
4. Após a conclusão da verificação, anexe a função ao grupo de identidades do Amazon Cognito que foi sinalizado como incompatível.

Você também pode usar ações de mitigação para:

- Aplicar `oPUBLISH_FINDINGS_TO_SNS`Ação de mitigação para implementar uma resposta personalizada em resposta à mensagem do Amazon SNS.

Para obter mais informações, consulte Ações de mitigação (p. 1045).

Gerenciar ou modificar coisas

Os seguintes exemplos deAWS IoTAs ações de API são usadas para gerenciar ou modificar coisas para que a permissão para executá-las não seja concedida a dispositivos que se conectem por meio de um grupo de identidades autenticadas do Amazon Cognito:

- `AddThingToThingGroup`
- `AttachThingPrincipal`
- `CreateThing`
- `DeleteThing`
- `DetachThingPrincipal`
- `ListThings`
- `ListThingsInThingGroup`
- `RegisterThing`
- `RemoveThingFromThingGroup`
- `UpdateThing`
- `UpdateThingGroupsForThing`

Qualquer função que conceda permissão para executar essas ações em até mesmo um único recurso é considerada não compatível.

Gerenciar não coisas

Os dispositivos que se conectam por meio de um grupo de identidades autenticadas do Amazon Cognito não devem receber permissão para executarAWS IoTAções de API diferentes daquelas discutidas nessas seções. Para gerenciar sua conta com um aplicativo que se conecta por meio de um grupo de identidades autenticadas do Amazon Cognito, crie um grupo de identidades separado não usado pelos dispositivos.

Ler dados administrativos das coisas

Os seguintes exemplos deAWS IoTAs ações de API são usadas para ler dados de coisas para que os dispositivos que se conectem por meio de um grupo de identidades autenticadas do Amazon Cognito tenham permissão para executá-las em apenas um conjunto limitado de coisas:

- `DescribeThing`
- `ListJobExecutionsForThing`
- `ListThingGroupsForThing`
- `ListThingPrincipals`

- incompatível:

```
arn:aws:iot:region:account-id:thing/*
```

  Isso permite que o dispositivo realize a ação especificada em qualquer coisa.
- compatível:

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

```
            "iot:DescribeThing",
            "iot:ListJobExecutionsForThing",
            "iot:ListThingGroupsForThing",
            "iot:ListThingPrincipals"
        ],
        "Resource": [
          "arn:aws:iot:region:account-id:/thing/MyThing"
        ]
      }
    ]
}
```

Isso permite que o dispositivo execute as ações especificadas em apenas uma coisa.

- compatível:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
          "iot:DescribeThing",
          "iot:ListJobExecutionsForThing",
          "iot:ListThingGroupsForThing",
          "iot:ListThingPrincipals"
      ],
      "Resource": [
        "arn:aws:iot:region:account-id:/thing/MyThing*"
      ]
    }
  ]
}
```

É compatível, pois, embora o recurso seja especificado com um caractere curinga (*), ele é precedido por uma string específica, e isso limita o conjunto de coisas acessadas àqueles com nomes que têm um determinado prefixo.

- incompatível:

```
arn:aws:iot:region:account-id:thing/*
```

Isso permite que o dispositivo realize a ação especificada em qualquer coisa.

- compatível:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
          "iot:DescribeThing",
          "iot:ListJobExecutionsForThing",
          "iot:ListThingGroupsForThing",
          "iot:ListThingPrincipals"
      ],
      "Resource": [
        "arn:aws:iot:region:account-id:/thing/MyThing"
      ]
    }
```

```
  ]
}
```

Isso permite que o dispositivo execute as ações especificadas em apenas uma coisa.

- compatível:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
          "iot:DescribeThing",
          "iot:ListJobExecutionsForThing",
          "iot:ListThingGroupsForThing",
          "iot:ListThingPrincipals"
      ],
      "Resource": [
        "arn:aws:iot:region:account-id:/thing/MyThing*"
      ]
    }
  ]
}
```

É compatível, pois, embora o recurso seja especificado com um caractere curinga (*), ele é precedido por uma string específica, e isso limita o conjunto de coisas acessadas àqueles com nomes que têm um determinado prefixo.

Assinar/publicar em tópicos do MQTT

As mensagens do MQTT são enviadas por meio do operador de mensagens da AWS IoT e são usadas pelos dispositivos para executar muitas ações diferentes, incluindo acessar e modificar o estado de shadow e o estado de execução de trabalhos. Uma política que concede permissão para um dispositivo se conectar, publicar ou assinar mensagens do MQTT deve restringir essas ações a recursos específicos da seguinte forma:

Conecte-se

- incompatível:

```
arn:aws:iot:region:account-id:client/*
```

O curinga * permite que qualquer dispositivo se conecte ao AWS IoT.

```
arn:aws:iot:region:account-id:client/${iot:ClientId}
```

A menos que `iot:Connection.Thing.IsAttached` seja definido como true nas chaves de condição, é equivalente ao curinga * no exemplo anterior.

- compatível:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [ "iot:Connect" ],
      "Resource": [
        "arn:aws:iot:region:account-id:client/${iot:Connection.Thing.ThingName}"
```

```
      ],
      "Condition": {
        "Bool": { "iot:Connection.Thing.IsAttached": "true" }
      }
    }
  ]
}
```

A especificação do recurso contém uma variável que corresponde ao nome do dispositivo usado para se conectar, e a declaração de condição restringe ainda mais a permissão, verificando se o certificado usado pelo cliente MQTT corresponde ao anexado à coisa com o nome usado.

Publicar

- incompatível:

```
arn:aws:iot:region:account-id:topic/$aws/things/*/shadow/update
```

Isso permite que o dispositivo atualize o shadow de qualquer dispositivo (* = todos os dispositivos).

```
arn:aws:iot:region:account-id:topic/$aws/things/*
```

Isso permite que o dispositivo leia/atualize/exclua o shadow de qualquer dispositivo.

- compatível:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [ "iot:Publish" ],
      "Resource": [
        "arn:aws:iot:region:account-id:topic/$aws/things/
${iot:Connection.Thing.ThingName}/shadow/*"
      ],
    }
  ]
}
```

A especificação do recurso contém um caractere curinga, mas apenas corresponde a qualquer tópico relacionado a shadow para o dispositivo cujo nome da coisa é usado para se conectar.

Assinar

- incompatível:

```
arn:aws:iot:region:account-id:topicfilter/$aws/things/*
```

Isso permite que o dispositivo assine tópicos de shadow ou de trabalho reservados para todos os dispositivos.

```
arn:aws:iot:region:account-id:topicfilter/$aws/things/#
```

O mesmo do exemplo anterior, mas usando o curinga #.

```
arn:aws:iot:region:account-id:topicfilter/$aws/things/+/shadow/update
```

Isso permite que o dispositivo veja as atualizações de shadow de qualquer dispositivo (+ = todos os dispositivos).

- compatível:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [ "iot:Subscribe" ],
      "Resource": [
        "arn:aws:iot:region:account-id:topicfilter/$aws/things/
${iot:Connection.Thing.ThingName}/shadow/*"
        "arn:aws:iot:region:account-id:topicfilter/$aws/things/
${iot:Connection.Thing.ThingName}/jobs/*"
      ],
    }
  ]
}
```

As especificações de recursos contêm caracteres curinga, mas eles apenas correspondem a qualquer tópico relacionado a shadow e a trabalho para o dispositivo cujo nome da coisa é usado para se conectar.

Receber

- compatível:

```
arn:aws:iot:region:account-id:topicfilter/$aws/things/*
```

Isso é compatível porque o dispositivo só pode receber mensagens de tópicos nos quais ele tem permissão para assinar.

Ler ou modificar dados de sombra ou trabalho

Uma política que concede permissão para um dispositivo executar uma ação de API para acessar ou modificar shadows de dispositivos ou dados de execução de trabalhos deve restringir essas ações a recursos específicos. Estas são as ações da API:

- `DeleteThingShadow`
- `GetThingShadow`
- `UpdateThingShadow`
- `DescribeJobExecution`
- `GetPendingJobExecutions`
- `StartNextPendingJobExecution`
- `UpdateJobExecution`

Exemplos

- incompatível:

```
arn:aws:iot:region:account-id:thing/*
```

Isso permite que o dispositivo realize a ação especificada em qualquer coisa.

- compatível:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
          "iot:DeleteThingShadow",
          "iot:GetThingShadow",
          "iot:UpdateThingShadow",
          "iot:DescribeJobExecution",
          "iot:GetPendingJobExecutions",
          "iot:StartNextPendingJobExecution",
          "iot:UpdateJobExecution"
      ],
      "Resource": [
        "arn:aws:iot:region:account-id:/thing/MyThing1",
        "arn:aws:iot:region:account-id:/thing/MyThing2"
      ]
    }
  ]
}
```

Isso permite que o dispositivo execute as ações específicas em apenas duas coisas.

AWS IoTPolíticas excessivamente permissivas

Uma política do AWS IoT concede permissões que são muito amplas ou irrestritas. Ela concede permissão para enviar ou receber mensagens MQTT para um amplo conjunto de dispositivos ou concede permissão para acessar ou modificar dados shadow e de execução de trabalhos para um amplo conjunto de dispositivos.

Em geral, uma política para um dispositivo deve conceder acesso a recursos associados a apenas esse dispositivo e nenhum outro ou a muito poucos dispositivos. Com algumas exceções, o uso de um curinga (por exemplo,"*") para especificar recursos em uma política é considerado muito amplo ou irrestrito.

Essa verificação aparece como`IOT_POLICY_OVERLY_PERMISSIVE_CHECK`na CLI e a API.

Gravidade: Critical

Detalhes

O código de motivo a seguir é retornado quando essa verificação encontra uma política incompatível do AWS IoT:

- ALLOWS_BROAD_ACCESS_TO_IOT_DATA_PLANE_ACTIONS

Por que isso importa?

Um certificado, uma identidade do Amazon Cognito ou um grupo de coisas com uma política excessivamente permissiva, se comprometidas, poderá afetar a segurança de toda a sua conta. Um invasor pode usar esse amplo acesso para ler ou modificar sombras, trabalhos ou execuções de trabalhos de todos os seus dispositivos. Ou um invasor pode usar um certificado comprometido para conectar dispositivos mal-intencionados ou ativar um ataque DDOS na rede.

Como corrigir

Siga estas etapas para corrigir todas as políticas que não estão em conformidade anexadas a coisas, grupos de coisas ou outras entidades:

1. Use CreatePolicyVersion para criar uma nova versão compatível da política. Defina o sinalizador `setAsDefault` como verdadeiro. (Isso torna essa nova versão operacional para todas as entidades que usam a política.)
2. Use ListTargetsForPolicy para obter uma lista de destinos (certificados, grupos de coisas) aos quais a política está anexada e determine quais dispositivos estão incluídos nos grupos ou que usam os certificados para se conectar.
3. Verifique se todos os dispositivos associados podem se conectar à AWS IoT. Se um dispositivo não conseguir se conectar, use SetPolicyVersion para reverter a política padrão para a versão anterior, revise-a e tente novamente.

Você pode usar ações de mitigação para:

- Aplicar a ação de mitigação `REPLACE_DEFAULT_POLICY_VERSION` em suas descobertas de auditoria para fazer essa mudança.
- Aplicar `oPUBLISH_FINDINGS_TO_SNS`Se você desejar implementar uma resposta personalizada em resposta à mensagem do Amazon SNS.

Para obter mais informações, consulte Ações de mitigação (p. 1045).

Use as variáveis da politica do AWS IoT (p. 336) para fazer referência dinâmica aos recursos do AWS IoT nas suas políticas.

Permissões MQTT

As mensagens do MQTT são enviadas por meio do agente de mensagens do AWS IoT e são usadas pelos dispositivos para executar muitas ações, incluindo acessar e modificar o estado de sombra e o estado de execução de trabalhos. Uma política que concede permissão para um dispositivo se conectar, publicar ou assinar mensagens do MQTT deve restringir essas ações a recursos específicos da seguinte forma:

Conecte-se

- incompatível:

```
arn:aws:iot:region:account-id:client/*
```

O curinga * permite que qualquer dispositivo se conecte ao AWS IoT.

```
arn:aws:iot:region:account-id:client/${iot:ClientId}
```

A não ser que `iot:Connection.Thing.IsAttached` seja definido como true nas chaves de condição, isso é equivalente ao curinga* como no exemplo anterior.

- compatível:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [ "iot:Connect" ],
      "Resource": [
        "arn:aws:iot:region:account-id:client/${iot:Connection.Thing.ThingName}"
      ],
      "Condition": {
        "Bool": { "iot:Connection.Thing.IsAttached": "true" }
      }
    }
  ]
```

```
}
```

A especificação de recurso contém uma variável que corresponde ao nome do dispositivo usado para se conectar. A declaração de condição restringe ainda mais a permissão, verificando se o certificado usado pelo cliente MQTT corresponde ao que é anexado à coisa com o nome usado.

Publicar

- incompatível:

```
arn:aws:iot:region:account-id:topic/$aws/things/*/shadow/update
```

Isso permite que o dispositivo atualize o shadow de qualquer dispositivo (* = todos os dispositivos).

```
arn:aws:iot:region:account-id:topic/$aws/things/*
```

Isso permite que o dispositivo leia, atualize ou exclua a sombra de qualquer dispositivo.

- compatível:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [ "iot:Publish" ],
      "Resource": [
        "arn:aws:iot:region:account-id:topic/$aws/things/
${iot:Connection.Thing.ThingName}/shadow/*"
      ],
    }
  ]
}
```

A especificação do recurso contém um caractere curinga, mas apenas corresponde a qualquer tópico relacionado a shadow para o dispositivo cujo nome da coisa é usado para se conectar.

Assinar

- incompatível:

```
arn:aws:iot:region:account-id:topicfilter/$aws/things/*
```

Isso permite que o dispositivo assine tópicos de shadow ou de trabalho reservados para todos os dispositivos.

```
arn:aws:iot:region:account-id:topicfilter/$aws/things/*
```

O mesmo do exemplo anterior, mas usando o curinga #.

```
arn:aws:iot:region:account-id:topicfilter/$aws/things/+/shadow/update
```

Isso permite que o dispositivo veja as atualizações de shadow de qualquer dispositivo (+ = todos os dispositivos).

- compatível:

```
{
  "Version": "2012-10-17",
```

```
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [ "iot:Subscribe" ],
      "Resource": [
        "arn:aws:iot:region:account-id:topicfilter/$aws/things/
${iot:Connection.Thing.ThingName}/shadow/*"
        "arn:aws:iot:region:account-id:topicfilter/$aws/things/
${iot:Connection.Thing.ThingName}/jobs/*"
      ],
    }
  ]
}
```

As especificações de recursos contêm caracteres curinga, mas eles apenas correspondem a qualquer tópico relacionado a shadow e a trabalho para o dispositivo cujo nome da coisa é usado para se conectar.

Receber

- compatível:

```
arn:aws:iot:region:account-id:topic/$aws/things/*
```

Isso é compatível porque o dispositivo só pode receber mensagens de tópicos nos quais ele tem permissão para assinar.

Permissões de sombra e trabalho

Uma política que concede permissão para um dispositivo executar uma ação de API para acessar ou modificar shadows de dispositivos ou dados de execução de trabalhos deve restringir essas ações a recursos específicos. Estas são as ações da API:

- `DeleteThingShadow`
- `GetThingShadow`
- `UpdateThingShadow`
- `DescribeJobExecution`
- `GetPendingJobExecutions`
- `StartNextPendingJobExecution`
- `UpdateJobExecution`

Exemplos

- incompatível:

```
arn:aws:iot:region:account-id:thing/*
```

Isso permite que o dispositivo realize a ação especificada em qualquer coisa.

- compatível:

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

```
            "iot:DeleteThingShadow",
            "iot:GetThingShadow",
            "iot:UpdateThingShadow",
            "iot:DescribeJobExecution",
            "iot:GetPendingJobExecutions",
            "iot:StartNextPendingJobExecution",
            "iot:UpdateJobExecution"
        ],
        "Resource": [
          "arn:aws:iot:region:account-id:/thing/MyThing1",
          "arn:aws:iot:region:account-id:/thing/MyThing2"
        ]
      }
    ]
}
```

Isso permite que o dispositivo execute as ações específicas em apenas duas coisas.

Alias de função excessivamente permissivo

AWS IoTalias de função fornece um mecanismo para que os dispositivos conectados sejam autenticados noAWS IoTusando certificados X.509 e, em seguida, obtenha curta duraçãoAWSCredenciais de uma função do IAM que está associada a umAWS IoTO alias da função. As permissões para essas credenciais devem ser definidas com escopo usando políticas de acesso com variáveis de contexto de autenticação. Se as políticas não estiverem configuradas corretamente, você ficará exposto a um escalonamento de ataque de privilégio. Essa verificação de auditoria garante que as credenciais temporárias fornecidas pelos aliases de função da AWS IoT não sejam excessivamente permissivas.

Essa verificação será acionada se uma das seguintes condições for encontrada:

- A política fornece permissões administrativas para todos os serviços usados no ano passado por esse alias de função (por exemplo, “iot:*”, “dynamodb:*”, “iam:*” e assim por diante).
- A política fornece amplo acesso a ações de metadados de coisa, acesso a ações restritas da AWS IoT ou amplo acesso a ações de plano de dados da AWS IoT.
- A política fornece acesso a serviços de auditoria de segurança como “iam”, “cloudtrail”, “guardduty”, “inspector” ou “trustedadvisor”.

Essa verificação aparece como`IOT_ROLE_ALIAS_OVERLY_PERMISSIVE_CHECK`na CLI e a API.

Gravidade: Critical

Detalhes

Os códigos de motivo a seguir são retornados quando essa verificação encontra uma política da IoT incompatível:

- ALLOWS_BROAD_ACCESS_TO_USED_SERVICES
- ALLOWS_ACCESS_TO_SECURITY_AUDITING_SERVICES
- ALLOWS_BROAD_ACCESS_TO_IOT_THING_ADMIN_READ_ACTIONS
- ALLOWS_ACCESS_TO_IOT_NON_THING_ADMIN_ACTIONS
- ALLOWS_ACCESS_TO_IOT_THING_ADMIN_WRITE_ACTIONS
- ALLOWS_BROAD_ACCESS_TO_IOT_DATA_PLANE_ACTIONS

Por que isso importa?

Ao limitar as permissões àquelas que são necessárias para que um dispositivo execute suas operações normais, você reduz os riscos para sua conta se um dispositivo estiver comprometido.

Como corrigir

Siga estas etapas para corrigir todas as políticas que não estão em conformidade anexadas a coisas, grupos de coisas ou outras entidades:

1. Siga as etapas em Autorizar chamadas diretas paraAWSUsar serviços doAWS IoT CoreProvedor de credenciais (p. 375) para aplicar uma política mais restritiva ao alias de função.

Você pode usar ações de mitigação para:

- Aplicar `oPUBLISH_FINDINGS_TO_SNS`Se você desejar implementar uma ação personalizada em resposta à mensagem do Amazon SNS.

Para obter mais informações, consulte Ações de mitigação (p. 1045).

O alias de função permite acesso a serviços não utilizados

AWS IoTalias de função fornece um mecanismo para que os dispositivos conectados sejam autenticados noAWS IoTusando certificados X.509 e, em seguida, obtenha curta duraçãoAWSCredenciais de uma função do IAM que está associada a umAWS IoTO alias da função. As permissões para essas credenciais devem ser definidas com escopo usando políticas de acesso com variáveis de contexto de autenticação. Se as políticas não estiverem configuradas corretamente, você ficará exposto a um escalonamento de ataque de privilégio. Essa verificação de auditoria garante que as credenciais temporárias fornecidas pelos aliases de função da AWS IoT não sejam excessivamente permissivas.

Essa verificação será acionada se o alias de função tiver acesso a serviços que não foram usados para o dispositivo AWS IoT no último ano. Por exemplo, os relatórios de auditoria se você tiver uma função do IAM vinculada ao alias de função que só tenha usadoAWS IoTno ano passado, mas a política anexada à função também concede permissão para`"iam:getRole"`e`"dynamodb:PutItem"`.

Essa verificação aparece como`IOT_ROLE_ALIAS_ALLOWS_ACCESS_TO_UNUSED_SERVICES_CHECK`na CLI e a API.

Gravidade: Médio

Detalhes

Os códigos de motivo a seguir são retornados quando essa verificação encontra uma política de AWS IoT não compatível:

- ALLOWS_ACCESS_TO_UNUSED_SERVICES

Por que isso importa?

Ao limitar as permissões aos serviços necessários para que um dispositivo execute suas operações normais, você reduz os riscos para sua conta se um dispositivo estiver comprometido.

Como corrigir

Siga estas etapas para corrigir todas as políticas que não estão em conformidade anexadas a coisas, grupos de coisas ou outras entidades:

1. Siga as etapas em Autorizar chamadas diretas paraAWSUsar serviços doAWS IoT CoreProvedor de credenciais (p. 375) para aplicar uma política mais restritiva ao alias de função.

Você pode usar ações de mitigação para:

- Aplicar `oPUBLISH_FINDINGS_TO_SNS`Se você desejar implementar uma ação personalizada em resposta à mensagem do Amazon SNS.

Para obter mais informações, consulte Ações de mitigação (p. 1045).

Certificado CA expirar

Um certificado da CA vai expirar dentro de 30 dias ou expirou.

Essa verificação aparece como`CA_CERTIFICATE_EXPIRING_CHECK`na CLI e a API.

Gravidade: Médio

Detalhes

Essa verificação se aplica a certificados da CA que estão ACTIVE ou PENDING_TRANSFER.

Os códigos de motivo a seguir são retornados quando essa verificação encontra um certificado da CA não compatível:

- CERTIFICATE_APPROACHING_EXPIRATION
- CERTIFICATE_PAST_EXPIRATION

Por que isso importa?

Um certificado da CA expirado não deve ser usado para assinar novos certificados de dispositivo.

Como corrigir

Consulte as práticas recomendadas de segurança para saber como proceder. Talvez você queira:

1. Registrar um novo certificado da CA com a AWS IoT.
2. Verificar se você pode assinar certificados de dispositivo usando os novos certificados da CA.
3. Use UpdateCACertificate para marcar o certificado antigo da CA como INACTIVE no AWS IoT. Você também pode usar ações de mitigação para fazer o seguinte:
   - Aplicar a ação de mitigação `UPDATE_CA_CERTIFICATE` em suas descobertas de auditoria para fazer essa mudança.
   - Aplicar `oPUBLISH_FINDINGS_TO_SNS`Se você desejar implementar uma resposta personalizada em resposta à mensagem do Amazon SNS.

   Para obter mais informações, consulte Ações de mitigação (p. 1045).

IDs do cliente MQTT conflitantes

Vários dispositivos se conectam usando o mesmo ID de cliente.

Essa verificação aparece como`CONFLICTING_CLIENT_IDS_CHECK`na CLI e a API.

Gravidade: Alto

Detalhes

Várias conexões foram feitas usando o mesmo ID de cliente, fazendo com que um dispositivo já conectado seja desconectado. A especificação do MQTT permite somente uma conexão ativa por ID de cliente. Por isso, quando outro dispositivo se conecta usando o mesmo ID de cliente, ele derruba a conexão do dispositivo anterior.

Quando executada como parte de uma auditoria sob demanda, essa verificação examina como os client IDs foram usados para se conectar durante os 31 dias anteriores ao início da auditoria. Para auditorias programadas, essa verificação analisa dados da última vez em que a auditoria foi executada até o momento em que essa instância da auditoria foi iniciada. Se você tiver tomado medidas para atenuar essa condição durante a verificação, observe quando as conexões/desconexões foram estabelecidas para determinar se o problema persiste.

Os códigos de motivo a seguir são retornados quando essa verificação encontra não compatibilidade:

- DUPLICATE_CLIENT_ID_ACROSS_CONNECTIONS

As descobertas retornadas por essa verificação também incluem o client ID usado para se conectar, IDs principais e tempos de desconexão. Os resultados mais recentes são listados primeiro.

Por que isso importa?

Os dispositivos com IDs conflitantes serão forçados a se reconectar constantemente, o que pode resultar em perda de mensagens ou incapacidade de um dispositivo de se conectar.

Isso pode indicar que um dispositivo ou credenciais de um dispositivo foram comprometidas e pode fazer parte de um ataque DDoS. Também é possível que os dispositivos não estejam configurados corretamente na conta ou que um dispositivo tenha uma conexão ruim e seja forçado a se reconectar várias vezes por minuto.

Como corrigir

Registre cada dispositivo como uma coisa exclusiva na AWS IoT e use o nome da coisa como o ID de cliente para se conectar. Ou use um UUID como o ID do cliente ao se conectar ao dispositivo por MQTT. Você também pode usar ações de mitigação para:

- Aplicar `oPUBLISH_FINDINGS_TO_SNS`Se você desejar implementar uma resposta personalizada em resposta à mensagem do Amazon SNS.

Para obter mais informações, consulte Ações de mitigação (p. 1045).

Certificado de dispositivo expirar

Um certificado de dispositivo vai expirar dentro de 30 dias ou expirou.

Essa verificação aparece como`DEVICE_CERTIFICATE_EXPIRING_CHECK`na CLI e a API.

Gravidade: Médio

Detalhes

Essa verificação se aplica a certificados de dispositivo que estão ACTIVE ou PENDING_TRANSFER.

Os códigos de motivo a seguir são retornados quando essa verificação encontra um certificado de dispositivo não compatível:

- CERTIFICATE_APPROACHING_EXPIRATION

- CERTIFICATE_PAST_EXPIRATION

Por que isso importa?

Um certificado de dispositivo não deve ser usado depois que ele expira.

Como corrigir

Consulte as práticas recomendadas de segurança para saber como proceder. Talvez você queira:

1. Provisione um novo certificado e o anexe ao dispositivo.
2. Verifique se o novo certificado é válido e se o dispositivo pode usá-lo para se conectar.
3. Use UpdateCertificate para marcar o certificado antigo como INACTIVE na AWS IoT. Você também pode usar ações de mitigação para:
   - Aplicar a ação de mitigação `UPDATE_DEVICE_CERTIFICATE` em suas descobertas de auditoria para fazer essa mudança.
   - Aplicar a ação de mitigação `ADD_THINGS_TO_THING_GROUP` para adicionar o dispositivo a um grupo, onde é possível executar uma ação sobre ele.
   - Aplicar `oPUBLISH_FINDINGS_TO_SNS`Se você desejar implementar uma resposta personalizada em resposta à mensagem do Amazon SNS.

   Para obter mais informações, consulte Ações de mitigação (p. 1045).
4. Desanexe o antigo certificado do dispositivo. (Consulte DetachThingPrincipal.)

Certificado de dispositivo revogado ainda ativo

Um certificado revogado do dispositivo ainda está ativo.

Essa verificação aparece como`REVOKED_DEVICE_CERTIFICATE_STILL_ACTIVE_CHECK`na CLI e a API.

Gravidade: Médio

Detalhes

Um certificado de dispositivo está na lista de revogação de certificados, mas ainda está ativo na AWS IoT.

Essa verificação se aplica a certificados de dispositivo que estão ACTIVE ou PENDING_TRANSFER.

Os códigos de motivo a seguir são retornados quando essa verificação encontra não compatibilidade:

- CERTIFICATE_REVOKED_BY_ISSUER

Por que isso importa?

Um dispositivo de certificado, em geral, é revogado porque foi comprometido. É possível que ele ainda não tenha sido revogado na AWS IoT devido a um erro ou uma supervisão.

Como corrigir

Verifique se o certificado de dispositivo não foi comprometido. Se foi, siga as práticas recomendadas de segurança para atenuar a situação. Talvez você queira:

1. Provisione um novo certificado para o dispositivo.
2. Verifique se o novo certificado é válido e se o dispositivo pode usá-lo para se conectar.

3. Use UpdateCertificate para marcar o certificado antigo como REVOKED no AWS IoT. Você também pode usar ações de mitigação para:
   - Aplicar a ação de mitigação `UPDATE_DEVICE_CERTIFICATE` em suas descobertas de auditoria para fazer essa mudança.
   - Aplicar a ação de mitigação `ADD_THINGS_TO_THING_GROUP` para adicionar o dispositivo a um grupo, onde é possível executar uma ação sobre ele.
   - Aplicar `oPUBLISH_FINDINGS_TO_SNS`Se você desejar implementar uma resposta personalizada em resposta à mensagem do Amazon SNS.

   Para obter mais informações, consulte Ações de mitigação (p. 1045).
4. Desanexe o antigo certificado do dispositivo. (Consulte DetachThingPrincipal.)

Como desabilitar o registro

AWS IoTOs logs não estão habilitados no Amazon CloudWatch. Verifica o registro em V1 e o V2 do.

Essa verificação aparece como`LOGGING_DISABLED_CHECK`na CLI e a API.

Gravidade: Baixo

Detalhes

Os códigos de motivo a seguir são retornados quando essa verificação encontra não compatibilidade:

- LOGGING_DISABLED

Por que isso importa?

AWS IoTlogs no CloudWatch fornecem visibilidade de comportamentos emAWS IoT, incluindo falhas de autenticação e conexões e desconexões inesperadas que podem indicar que um dispositivo foi comprometido.

Como corrigir

Habilitar oAWS IoTfaz logs no CloudWatch. Consulte Ferramentas de monitoramento (p. 420). Você também pode usar ações de mitigação para:

- Aplicar a ação de mitigação `ENABLE_IOT_LOGGING` em suas descobertas de auditoria para fazer essa mudança.
- Aplicar `oPUBLISH_FINDINGS_TO_SNS`Se você desejar implementar uma resposta personalizada em resposta à mensagem do Amazon SNS.

Para obter mais informações, consulte Ações de mitigação (p. 1045).

Comandos de auditoria

Gerenciar configurações de auditoria

Use `UpdateAccountAuditConfiguration` para definir as configurações de auditoria de sua conta. Esse comando permite habilitar as verificações que devem estar disponíveis para auditorias, configurar notificações opcionais e configurar permissões.

Verifique essas configurações com `DescribeAccountAuditConfiguration`.

Use `DeleteAccountAuditConfiguration` para excluir suas configurações de auditoria. Isso restaura todos os valores padrão e desativa efetivamente as auditorias, pois todas as verificações estão desativadas por padrão.

UpdateAccountAuditConfiguration

Configura ou reconfigura o as configurações de auditoria do Device Defender para essa conta. As configurações incluem como as notificações de auditoria são enviadas e quais verificações de auditoria estão ativadas ou desativadas.

Resumo

```
aws iot  update-account-audit-configuration \
    [--role-arn <value>] \
    [--audit-notification-target-configurations <value>] \
    [--audit-check-configurations <value>]  \
    [--cli-input-json <value>] \
    [--generate-cli-skeleton]
```

Formato de `cli-input-json`

```
{
  "roleArn": "string",
  "auditNotificationTargetConfigurations": {
    "string": {
      "targetArn": "string",
      "roleArn": "string",
      "enabled": "boolean"
    }
  },
  "auditCheckConfigurations": {
    "string": {
      "enabled": "boolean"
    }
  }
}
```

Campos **`cli-input-json`**

Name (Nome)	Type	Descrição
roleArn	string comprimento - máx.: 2048 mín.: 20	O ARN da função que concede permissão à AWS IoT para acessar informações sobre os dispositivos, políticas, certificados e outros itens ao executar uma auditoria.
auditNotificationTargetConfigurations	mapear	Informações sobre os destinos para os quais as notificações de auditoria são enviadas.
targetArn	string	O ARN do destino (tópico do SNS) para o qual as notificações de auditoria são enviadas.
roleArn	string	O ARN da função que concede permissão para enviar notificações ao destino.

Name (Nome)	Type	Descrição
	comprimento - máx.: 2048 mín.: 20	
enabled	booliano	Verdadeiro se as notificações para o destino estão ativadas.
auditCheckConfigurations	mapear	Especifica quais verificações de auditoria são ativadas ou desativadas para essa conta. Use `DescribeAccountAuditConfiguration` para consultar a lista de todas as verificações, incluindo aquelas ativadas atualmente. A coleta de alguns dados pode começar imediatamente quando determinadas verificações forem ativadas. Quando uma verificação for desativada, todos os dados coletados até o momento em relação à verificação são excluídos. Não é possível desativar uma verificação se ela for usada por qualquer auditoria programada. Primeiro é necessário excluir a verificação da auditoria programada ou excluir a auditoria em si. Na primeira chamada ao `UpdateAccountAuditConfiguration`, esse parâmetro é necessário e deve especificar pelo menos uma verificação ativada.
enabled	booliano	Verdadeiro se essa verificação de auditoria está ativada para essa conta.

Resultado

Nenhum

Erros

`InvalidRequestException`

O conteúdo da solicitação era inválido.

`ThrottlingException`

A taxa excede o limite.

`InternalFailureException`

Ocorreu um erro inesperado.

DescribeAccountAuditConfiguration

Obtém informações sobre as configurações de auditoria do Device Defender para esta conta. As configurações incluem como as notificações de auditoria são enviadas e quais verificações de auditoria estão ativadas ou desativadas.

Resumo

```
aws iot  describe-account-audit-configuration  \
    [--cli-input-json <value>] \
    [--generate-cli-skeleton]
```

Formato de `cli-input-json`

```
{
}
```

Resultado

```
{
  "roleArn": "string",
  "auditNotificationTargetConfigurations": {
    "string": {
      "targetArn": "string",
      "roleArn": "string",
      "enabled": "boolean"
    }
  },
  "auditCheckConfigurations": {
    "string": {
      "enabled": "boolean"
    }
  }
}
```

Campos de saída da CLI

Name (Nome)	Type	Descrição
roleArn	string comprimento - máx.: 2048 mín.: 20	O ARN da função que concede permissão à AWS IoT para acessar informações sobre os dispositivos, políticas, certificados e outros itens ao executar uma auditoria. Na primeira chamada ao `UpdateAccountAuditConfiguration`, esse parâmetro é obrigatório.
auditNotificationTargetConfigurations	mapear	Informações sobre os destinos para os quais as notificações são enviadas para essa conta.
targetArn	string	O ARN do destino (tópico do SNS) para o qual as notificações de auditoria são enviadas.

Name (Nome)	Type	Descrição
roleArn	string comprimento - máx.: 2048 mín.: 20	O ARN da função que concede permissão para enviar notificações ao destino.
enabled	booliano	Verdadeiro se as notificações para o destino estão ativadas.
auditCheckConfigurations	mapear	Quais verificações de auditoria são ativadas ou desativadas para essa conta.
enabled	booliano	Verdadeiro se essa verificação de auditoria está ativada para essa conta.

Erros

`ThrottlingException`

A taxa excede o limite.

`InternalFailureException`

Ocorreu um erro inesperado.

DeleteAccountAuditConfiguration

Restaura as configurações padrão para auditorias do Device Defender para essa conta. Qualquer dado de configuração inserido é excluído e todas as verificações de auditoria são desativadas.

Resumo

```
aws iot  delete-account-audit-configuration \
    [--delete-scheduled-audits | --no-delete-scheduled-audits]  \
    [--cli-input-json <value>] \
    [--generate-cli-skeleton]
```

Formato de `cli-input-json`

```
{
  "deleteScheduledAudits": "boolean"
}
```

Campos **`cli-input-json`**

Name (Nome)	Type	Descrição
deleteScheduledAudits	booliano	Se verdadeiro, todas as auditorias programadas são excluídas.

Resultado

Nenhum

Erros

`InvalidRequestException`

O conteúdo da solicitação era inválido.

`ResourceNotFoundException`

O recurso especificado não existe.

`ThrottlingException`

A taxa excede o limite.

`InternalFailureException`

Ocorreu um erro inesperado.

Programar auditorias

Use `CreateScheduledAudit` para criar uma ou mais auditorias programadas. Este comando permite especificar as verificações que você deseja executar durante uma auditoria e a frequência com que essa auditoria deve ser executada.

Acompanhe suas auditorias programadas com `ListScheduledAudits` e `DescribeScheduledAudit`.

Altere uma auditoria programada existente com `UpdateScheduledAudit` ou a exclua com `DeleteScheduledAudit`.

CreateScheduledAudit

Cria uma auditoria programada que é executada em um intervalo de tempo especificado.

Resumo

```
aws iot  create-scheduled-audit \
    --frequency <value> \
    [--day-of-month <value>] \
    [--day-of-week <value>] \
    --target-check-names <value> \
    [--tags <value>] \
    --scheduled-audit-name <value>  \
    [--cli-input-json <value>] \
    [--generate-cli-skeleton]
```

Formato de `cli-input-json`

```
{
  "frequency": "string",
  "dayOfMonth": "string",
  "dayOfWeek": "string",
  "targetCheckNames": [
    "string"
  ],
  "tags": [
    {
      "Key": "string",
```

```
      "Value": "string"
    }
  ],
  "scheduledAuditName": "string"
}
```

Campos **`cli-input-json`**

Name (Nome)	Type	Descrição
frequency	string	A frequência com que a auditoria programada ocorre. Pode ser DAILY, WEEKLY, BIWEEKLY ou MONTHLY. A hora de início real de cada auditoria é determinada pelo sistema. enum: DIÁRIO | SEMANAL | QUINZENAL | MENSAL
dayOfMonth	string padrão: ^ ([1-9] | [0-9] |3 [01]) $\| ^ULAR$	O dia do mês em que a auditoria programada ocorre. Pode ser 1 a 31 ou LAST. Este campo é obrigatório se o parâmetro `frequency` for definido como MONTHLY. Se os dias 29 a 31 forem especificados e o mês não tiver essa quantidade de dias, a auditoria ocorre no LAST dia do mês.
dayOfWeek	string	O dia da semana em que a auditoria programada ocorre. Pode ser SUN, MON, TUE,WED, THU, FRI, ou SAT. Este campo é obrigatório se o parâmetro `frequency` for definido como WEEKLY or BIWEEKLY. enum: SOL | SEG | TER | QUA | QUI | SEX | SAT
targetCheckNames	list membro: AuditCheckName	Quais verificações são realizadas durante a auditoria programada. As verificações devem estar ativadas na sua conta. (Use `DescribeAccountAuditConfiguration` para consultar a lista de todas as verificações, incluindo as que estão ativadas ou `UpdateAccountAuditConfiguration` para selecionar quais verificações serão ativadas.)
tags	list membro: Tag classe java: java.util.List	Metadados que podem ser usados para gerenciar a auditoria programada.

Name (Nome)	Type	Descrição
Chave	string	A chave da tag.
Valor	string	O valor da tag.
scheduledAuditName	string comprimento - máx.: 128 mín.: 1 padrão: [a-zA-Z0-9_-]+	O nome que deseja dar à auditoria programada. (Máximo de 128 caracteres)

Resultado

```
{
  "scheduledAuditArn": "string"
}
```

Campos de saída da CLI

Name (Nome)	Type	Descrição
scheduledAuditArn	string	O ARN da auditoria programada.

Erros

`InvalidRequestException`

O conteúdo da solicitação era inválido.

`ThrottlingException`

A taxa excede o limite.

`InternalFailureException`

Ocorreu um erro inesperado.

`LimitExceededException`

Um limite foi excedido.

ListScheduledAudits

Lista todas as auditorias programadas.

Resumo

```
aws iot  list-scheduled-audits \
    [--next-token <value>] \
    [--max-results <value>]  \
    [--cli-input-json <value>] \
    [--generate-cli-skeleton]
```

Formato de `cli-input-json`

```
{
```

```
  "nextToken": "string",
  "maxResults": "integer"
}
```

Campos **cli-input-json**

Name (Nome)	Type	Descrição
nextToken	string	O token para o próximo conjunto de resultados.
maxResults	integer range- máx.: 250, mín.: 1	O número máximo de resultados a serem retornados ao mesmo tempo. O padrão é 25.

Resultado

```
{
  "scheduledAudits": [
    {
      "scheduledAuditName": "string",
      "scheduledAuditArn": "string",
      "frequency": "string",
      "dayOfMonth": "string",
      "dayOfWeek": "string"
    }
  ],
  "nextToken": "string"
}
```

Campos de saída da CLI

Name (Nome)	Type	Descrição
scheduledAudits	list membro: ScheduledAuditMetadata classe java: java.util.List	A lista de auditorias programadas.
scheduledAuditName	string comprimento - máx.: 128 mín.: 1 padrão: [a-zA-Z0-9_-]+	O nome da auditoria programada.
scheduledAuditArn	string	O ARN da auditoria programada.
frequency	string	A frequência com que a auditoria programada ocorre. enum: DIÁRIO | SEMANAL | QUINZENAL | MENSAL
dayOfMonth	string padrão: ^ ([1-9] | [0-9] |3 [01]) $\| ^ULAR$	O dia do mês em que a auditoria programada é executada (se `frequency` for MONTHLY). Se os dias 29 a 31 forem

Name (Nome)	Type	Descrição
		especificados e o mês não tiver essa quantidade de dias, a auditoria ocorre no LAST dia do mês.
dayOfWeek	string	O dia da semana em que a auditoria programada é executada (se `frequency` for WEEKLY ou BIWEEKLY). enum: SOL | SEG | TER | QUA | QUI | SEX | SAT
nextToken	string	Um token que pode ser usado para recuperar o próximo conjunto de resultados ou `null` se não houver mais resultados.

Erros

`InvalidRequestException`

O conteúdo da solicitação era inválido.

`ThrottlingException`

A taxa excede o limite.

`InternalFailureException`

Ocorreu um erro inesperado.

DescribeScheduledAudit

Obtém informações sobre uma auditoria programada.

Resumo

```
aws iot  describe-scheduled-audit \
    --scheduled-audit-name <value>  \
    [--cli-input-json <value>] \
    [--generate-cli-skeleton]
```

Formato de `cli-input-json`

```
{
  "scheduledAuditName": "string"
}
```

Campos **`cli-input-json`**

Name (Nome)	Type	Descrição
scheduledAuditName	string comprimento - máx.: 128 mín.: 1	O nome da auditoria programada da qual deseja obter informações.

Name (Nome)	Type	Descrição
	padrão: [a-zA-Z0-9_-]+	

Resultado

```
{
  "frequency": "string",
  "dayOfMonth": "string",
  "dayOfWeek": "string",
  "targetCheckNames": [
    "string"
  ],
  "scheduledAuditName": "string",
  "scheduledAuditArn": "string"
}
```

Campos de saída da CLI

Name (Nome)	Type	Descrição
frequency	string	A frequência com que a auditoria programada ocorre. DAILY, WEEKLY, BIWEEKLY, ou MONTHLY. A hora de início real de cada auditoria é determinada pelo sistema. enum: DIÁRIO | SEMANAL | QUINZENAL | MENSAL
dayOfMonth	string padrão: ^ ([1-9] | [0-9] |3 [01]) $\| ^ULAR$	O dia do mês em que a auditoria programada ocorre. Pode ser 1 a 31 ou LAST. Se os dias 29 a 31 forem especificados e o mês não tiver essa quantidade de dias, a auditoria ocorre no LAST dia do mês.
dayOfWeek	string	O dia da semana em que a auditoria programada ocorre. SUN, MON, TUE, WED, THU, FRI, ou SAT. enum: SOL | SEG | TER | QUA | QUI | SEX | SAT
targetCheckNames	list membro: AuditCheckName	Quais verificações são realizadas durante a auditoria programada. As verificações devem estar ativadas na sua conta. (Use `DescribeAccountAuditConfiguration` para consultar a lista de todas as verificações, incluindo as que estão ativadas ou use `UpdateAccountAuditConfiguration` para selecionar quais verificações serão ativadas.)

Name (Nome)	Type	Descrição
scheduledAuditName	string comprimento - máx.: 128 mín.: 1 padrão: [a-zA-Z0-9_-]+	O nome da auditoria programada.
scheduledAuditArn	string	O ARN da auditoria programada.

Erros

`InvalidRequestException`

O conteúdo da solicitação era inválido.

`ResourceNotFoundException`

O recurso especificado não existe.

`ThrottlingException`

A taxa excede o limite.

`InternalFailureException`

Ocorreu um erro inesperado.

UpdateScheduledAudit

Atualiza uma auditoria programada, incluindo quais verificações são realizadas e a frequência com que a auditoria ocorre.

Resumo

```
aws iot  update-scheduled-audit \
    [--frequency <value>] \
    [--day-of-month <value>] \
    [--day-of-week <value>] \
    [--target-check-names <value>] \
    --scheduled-audit-name <value>  \
    [--cli-input-json <value>] \
    [--generate-cli-skeleton]
```

Formato de `cli-input-json`

```
{
  "frequency": "string",
  "dayOfMonth": "string",
  "dayOfWeek": "string",
  "targetCheckNames": [
    "string"
  ],
  "scheduledAuditName": "string"
}
```

Campos **`cli-input-json`**

Name (Nome)	Type	Descrição
frequency	string	A frequência com que a auditoria programada ocorre. Pode ser DAILY, WEEKLY, BIWEEKLY ou MONTHLY. A hora de início real de cada auditoria é determinada pelo sistema. enum: DIÁRIO | SEMANAL | QUINZENAL | MENSAL
dayOfMonth	string padrão: ^ ([1-9] | [0-9] |3 [01]) $\| ^ULAR$	O dia do mês em que a auditoria programada ocorre. Pode ser 1 a 31 ou LAST. Este campo é obrigatório se o parâmetro `frequency` for definido como MONTHLY. Se os dias 29 a 31 forem especificados e o mês não tiver essa quantidade de dias, a auditoria ocorre no LAST dia do mês.
dayOfWeek	string	O dia da semana em que a auditoria programada ocorre. Pode ser SUN, MON, TUE, WED, THU, FRI, ou SAT. Este campo é obrigatório se o parâmetro `frequency` for definido como WEEKLY or BIWEEKLY. enum: SOL | SEG | TER | QUA | QUI | SEX | SAT
targetCheckNames	list membro: AuditCheckName	Quais verificações são realizadas durante a auditoria programada. As verificações devem estar ativadas na sua conta. (Use `DescribeAccountAuditConfiguration` para consultar a lista de todas as verificações, incluindo as que estão ativadas ou use `UpdateAccountAuditConfiguration` para selecionar quais verificações serão ativadas.)
scheduledAuditName	string comprimento - máx.: 128 mín.: 1 padrão: [a-zA-Z0-9_-]+	O nome da auditoria programada. (Máximo de 128 caracteres)

Resultado

```
{
```

```
  "scheduledAuditArn": "string"
}
```

Campos de saída da CLI

Name (Nome)	Type	Descrição
scheduledAuditArn	string	O ARN da auditoria programada.

Erros

`InvalidRequestException`

O conteúdo da solicitação era inválido.

`ResourceNotFoundException`

O recurso especificado não existe.

`ThrottlingException`

A taxa excede o limite.

`InternalFailureException`

Ocorreu um erro inesperado.

DeleteScheduledAudit

Exclui uma auditoria programada.

Resumo

```
aws iot  delete-scheduled-audit \
    --scheduled-audit-name <value>  \
    [--cli-input-json <value>] \
    [--generate-cli-skeleton]
```

Formato de `cli-input-json`

```
{
  "scheduledAuditName": "string"
}
```

Campos **`cli-input-json`**

Name (Nome)	Type	Descrição
scheduledAuditName	string comprimento - máx.: 128 mín.: 1 padrão: [a-zA-Z0-9_-]+	O nome da auditoria programada que deseja excluir.

Resultado

Nenhum

Erros

`InvalidRequestException`

O conteúdo da solicitação era inválido.

`ResourceNotFoundException`

O recurso especificado não existe.

`ThrottlingException`

A taxa excede o limite.

`InternalFailureException`

Ocorreu um erro inesperado.

Executar uma auditoria sob demanda

Use `StartOnDemandAuditTask` para especificar as verificações que você deseja executar e iniciar uma auditoria com execução imediata.

StartOnDemandAuditTask

Iniciar uma auditoria do Device Defender sob demanda.

Resumo

```
aws iot  start-on-demand-audit-task \
    --target-check-names <value>  \
    [--cli-input-json <value>] \
    [--generate-cli-skeleton]
```

Formato de `cli-input-json`

```
{
  "targetCheckNames": [
    "string"
  ]
}
```

Campos **`cli-input-json`**

Name (Nome)	Type	Descrição
targetCheckNames	list membro: AuditCheckName	Quais verificações são realizadas durante a auditoria. As verificações especificadas devem estar ativadas na sua conta ou ocorrerá uma exceção. Use `DescribeAccountAuditConfiguration` para consultar a lista de todas as verificações, incluindo as que estão ativadas ou `UpdateAccountAuditConfiguration` para selecionar quais verificações serão ativadas.

Resultado

```
{
  "taskId": "string"
}
```

Campos de saída da CLI

Name (Nome)	Type	Descrição
taskId	string comprimento - máx.: 40 mín.: 1 padrão: [a-zA-Z0-9-]+	O ID da auditoria sob demanda iniciada.

Erros

`InvalidRequestException`

O conteúdo da solicitação era inválido.

`ThrottlingException`

A taxa excede o limite.

`InternalFailureException`

Ocorreu um erro inesperado.

`LimitExceededException`

Um limite foi excedido.

Gerenciar instâncias de auditoria

Use `DescribeAuditTask` para obter informações sobre uma instância de auditoria específica. Se ela já foi executada, os resultados incluem quais verificações falharam e quais foram aprovadas, aquelas que o sistema não foi capaz de concluir e, se a auditoria ainda estiver em andamento, aquelas em que ainda está trabalhando.

Use `ListAuditTasks` para encontrar as auditorias executadas durante um intervalo de tempo especificado.

Use `CancelAuditTask` para interromper uma auditoria em andamento.

DescribeAuditTask

Obtém informações sobre uma auditoria do Device Defender.

Resumo

```
aws iot  describe-audit-task \
    --task-id <value>  \
    [--cli-input-json <value>] \
    [--generate-cli-skeleton]
```

Formato de `cli-input-json`

```
{
  "taskId": "string"
}
```

Campos **`cli-input-json`**

Name (Nome)	Type	Descrição
taskId	string comprimento - máx.: 40 mín.: 1 padrão: [a-zA-Z0-9-]+	O ID da auditoria programada da qual deseja obter informações.

Resultado

```
{
  "taskStatus": "string",
  "taskType": "string",
  "taskStartTime": "timestamp",
  "taskStatistics": {
    "totalChecks": "integer",
    "inProgressChecks": "integer",
    "waitingForDataCollectionChecks": "integer",
    "compliantChecks": "integer",
    "nonCompliantChecks": "integer",
    "failedChecks": "integer",
    "canceledChecks": "integer"
  },
  "scheduledAuditName": "string",
  "auditDetails": {
    "string": {
      "checkRunStatus": "string",
      "checkCompliant": "boolean",
      "totalResourcesCount": "long",
      "nonCompliantResourcesCount": "long",
      "errorCode": "string",
      "message": "string"
    }
  }
}
```

Campos de saída da CLI

Name (Nome)	Type	Descrição
taskStatus	string	O status da auditoria: IN_PROGRESS, COMPLETED, FAILED, ou CANCELED. enum: IN_PROGRESS | CONCLUÍDO | FALHOU | CANCELADO
taskTipo	string	O tipo de auditoria: ON_DEMAND_AUDIT_TASK ou SCHEDULED_AUDIT_TASK. enum: ON_DEMAND_AUDIT_TASK | SCHEDULED_AUDIT_TASK
taskStartTime	timestamp	A hora em que a auditoria foi iniciada.

Name (Nome)	Type	Descrição
taskStatistics	TaskStatistics	Informações estatísticas sobre a auditoria.
totalChecks	integer	O número de verificações nesta auditoria.
inProgressChecks	integer	O número de verificações em andamento.
waitingForDataCollectionChecks	integer	O número de verificações aguardando a coleta de dados.
compliantChecks	integer	O número de verificações que encontraram recursos compatíveis.
nonCompliantChecks	integer	O número de verificações que encontraram recursos incompatíveis.
failedChecks	integer	O número de verificações.
canceledChecks	integer	O número de verificações que não foram executadas porque a auditoria foi cancelada.
scheduledAuditName	string comprimento - máx.: 128 mín.: 1 padrão: [a-zA-Z0-9_-]+	O nome da auditoria programada (somente se a auditoria for uma auditoria programada).
auditDetails	mapear	Informações detalhadas sobre cada verificação realizada durante essa auditoria.
checkRunStatus	string	O status de conclusão dessa verificação: IN_PROGRESS, WAITING_FOR_DATA_COLLECTION, CANCELED, COMPLETED_COMPLIANT, COMPLETED_NON_COMPLIANT, ou FAILED. enum: IN_PROGRESS | WAITING_FOR_DATA_COLLECTION | CANCELADO | COMPLETED_COMPLIANT | COMPLETED_NON_COMPLIANT | FALHOU
checkCompliant	booliano	Verdadeiro se a verificação foi concluída e encontrou todos os recursos compatíveis.
totalResourcesCount	longo	O número de recursos nos quais a verificação foi realizada.

Name (Nome)	Type	Descrição
nonCompliantResourcesCount	longo	O número de recursos que a verificação identificou como incompatíveis.
errorCode	string	O código de qualquer erro encontrado ao executar essa verificação durante a auditoria. INSUFFICIENT_PERMISSIONS ou AUDIT_CHECK_DISABLED.
mensagem	string comprimento - máx.: 2048	A mensagem associada a qualquer erro encontrado ao executar essa verificação durante a auditoria.

Erros

`InvalidRequestException`

O conteúdo da solicitação era inválido.

`ResourceNotFoundException`

O recurso especificado não existe.

`ThrottlingException`

A taxa excede o limite.

`InternalFailureException`

Ocorreu um erro inesperado.

ListAuditTasks

Lista as auditorias do Device Defender realizadas durante um determinado período.

Resumo

```
aws iot  list-audit-tasks \
    --start-time <value> \
    --end-time <value> \
    [--task-type <value>] \
    [--task-status <value>] \
    [--next-token <value>] \
    [--max-results <value>]  \
    [--cli-input-json <value>] \
    [--generate-cli-skeleton]
```

Formato de `cli-input-json`

```
{
  "startTime": "timestamp",
  "endTime": "timestamp",
  "taskType": "string",
  "taskStatus": "string",
```

```
  "nextToken": "string",
  "maxResults": "integer"
}
```

Campos **`cli-input-json`**

Name (Nome)	Type	Descrição
startTime	timestamp	O início do período. As informações de auditoria são mantidas durante um período limitado (180 dias). Solicitar uma hora de início anterior ao que está mantido resulta em um `InvalidRequestException`.
endTime	timestamp	O término do período.
taskTipo	string	Um filtro para limitar a saída para o tipo de auditoria especificado: pode ser ON_DEMAND_AUDIT_TASK ou SCHEDULED__AUDIT_TASK. enum: ON_DEMAND_AUDIT_TASK | SCHEDULED_AUDIT_TASK
taskStatus	string	Um filtro para limitar a saída para auditorias com o status de conclusão especificado: pode ser IN_PROGRESS, COMPLETED, FAILED, ou CANCELED. enum: IN_PROGRESS | CONCLUÍDO | FALHOU | CANCELADO
nextToken	string	O token para o próximo conjunto de resultados.
maxResults	integer range- máx.: 250, mín.: 1	O número máximo de resultados a serem retornados ao mesmo tempo. O padrão é 25.

Resultado

```
{
  "tasks": [
    {
      "taskId": "string",
      "taskStatus": "string",
      "taskType": "string"
    }
  ],
  "nextToken": "string"
}
```

Campos de saída da CLI

Name (Nome)	Type	Descrição
tarefas	list membro: AuditTaskMetadata classe java: java.util.List	As auditorias realizadas durante o período especificado.
taskId	string comprimento - máx.: 40 mín.: 1 padrão: [a-zA-Z0-9-]+	O ID dessa auditoria.
taskStatus	string	O status dessa auditoria: IN_PROGRESS, COMPLETED, FAILED, ou CANCELED. enum: IN_PROGRESS | CONCLUÍDO | FALHOU | CANCELADO
taskTipo	string	O tipo dessa auditoria: ON_DEMAND_AUDIT_TASK ou SCHEDULED_AUDIT_TASK. enum: ON_DEMAND_AUDIT_TASK | SCHEDULED_AUDIT_TASK
nextToken	string	Um token que pode ser usado para recuperar o próximo conjunto de resultados ou `null` se não houver resultados adicionais.

Erros

`InvalidRequestException`

O conteúdo da solicitação era inválido.

`ThrottlingException`

A taxa excede o limite.

`InternalFailureException`

Ocorreu um erro inesperado.

CancelAuditTask

Cancela uma auditoria que está em andamento. A auditoria pode ser programada ou sob demanda. Se a auditoria não está em andamento, uma `InvalidRequestException` ocorre.

Resumo

```
aws iot  cancel-audit-task \
```

```
    --task-id <value>  \
    [--cli-input-json <value>] \
    [--generate-cli-skeleton]
```

Formato de `cli-input-json`

```
{
  "taskId": "string"
}
```

Campos **`cli-input-json`**

Name (Nome)	Type	Descrição
taskId	string comprimento - máx.: 40 mín.: 1 padrão: [a-zA-Z0-9-]+	O ID da auditoria que deseja cancelar. Só é possível cancelar uma auditoria que está IN_PROGRESS.

Resultado

Nenhum

Erros

`ResourceNotFoundException`

O recurso especificado não existe.

`InvalidRequestException`

O conteúdo da solicitação era inválido.

`ThrottlingException`

A taxa excede o limite.

`InternalFailureException`

Ocorreu um erro inesperado.

Verificar os resultados da auditoria

Use `ListAuditFindings` para ver os resultados de uma auditoria. É possível filtrar os resultados pelo tipo de verificação, um recurso específico ou a data da auditoria. Você pode usar essas informações para reduzir os problemas que foram encontrados.

Você pode definir ações de mitigação e aplicá-las às descobertas de sua auditoria. Para obter mais informações, consulte Ações de mitigação (p. 1045).

ListAuditFindings

Lista as descobertas (resultados) de uma auditoria do Device Defender ou das auditorias realizadas durante um período especificado. (As descobertas são mantidas durante 180 dias.)

Resumo

```
aws iot  list-audit-findings \
    [--task-id <value>] \
    [--check-name <value>] \
    [--resource-identifier <value>] \
    [--max-results <value>] \
    [--next-token <value>] \
    [--start-time <value>] \
    [--end-time <value>]  \
    [--cli-input-json <value>] \
    [--generate-cli-skeleton]
```

Formato de `cli-input-json`

```
{
  "taskId": "string",
  "checkName": "string",
  "resourceIdentifier": {
    "deviceCertificateId": "string",
    "caCertificateId": "string",
    "cognitoIdentityPoolId": "string",
    "clientId": "string",
    "policyVersionIdentifier": {
      "policyName": "string",
      "policyVersionId": "string"
    },

    "roleAliasArn": "string",
    "account": "string"
  },
  "maxResults": "integer",
  "nextToken": "string",
  "startTime": "timestamp",
  "endTime": "timestamp"
}
```

Campos **`cli-input-json`**

Name (Nome)	Type	Descrição
taskId	string comprimento - máx.: 40 mín.: 1 padrão: [a-zA-Z0-9-]+	Um filtro para limitar os resultados para a auditoria com o ID especificado. É necessário especificar o taskId, ou startTime e endTime, mas não ambos.
checkName	string	Um filtro para limitar os resultados para as descobertas da verificação de auditoria especificada.
resourceIdentifier	ResourceIdentifier	As informações que identificam o recurso incompatível.
deviceCertificateId	string comprimento - máx.: 64 mín.: 64 padrão: (0x)?[a-fA-F0-9]+	O ID do certificado anexado ao recurso.
caCertificateId	string	O ID do certificado CA usado para autorizar o certificado.

Name (Nome)	Type	Descrição
	comprimento - máx.: 64 mín.: 64 padrão: (0x)?[a-fA-F0-9]+	
cognitoIdentityPoolId	string	O ID do grupo de identidades do Amazon Cognito.
clientId	string	O ID do cliente.
policyVersionIdentifier	PolicyVersionIdentifier	A versão da política associada ao recurso.
policyName	string comprimento - máx.: 128 mín.: 1 padrão: [w+=,.@-]+	O nome da política.
policyVersionId	string padrão: [0-9] +	O ID da versão da política associada ao recurso.
roleAliasArn	string	O ARN do alias de função que tem ações excessivamente permissivas. comprimento - máx.: 2048 mín.: 1
account	string comprimento - máx.: 12 mín.: 12 padrão: [0-9] +	A conta com a qual o recurso está associado.
maxResults	integer range- máx.: 250, mín.: 1	O número máximo de resultados a serem retornados ao mesmo tempo. O padrão é 25.
nextToken	string	O token para o próximo conjunto de resultados.
startTime	timestamp	Um filtro para limitar os resultados para aqueles encontrados após o tempo especificado. É necessário especificar startTime e endTime, ou o taskId, mas não ambos.
endTime	timestamp	Um filtro para limitar os resultados para aqueles encontrados antes do tempo especificado. É necessário especificar startTime e endTime, ou o taskId, mas não ambos.

Resultado

```
{
```

```
  "findings": [
    {
      "taskId": "string",
      "checkName": "string",
      "taskStartTime": "timestamp",
      "findingTime": "timestamp",
      "severity": "string",
      "nonCompliantResource": {
        "resourceType": "string",
        "resourceIdentifier": {
          "deviceCertificateId": "string",
          "caCertificateId": "string",
          "cognitoIdentityPoolId": "string",
          "clientId": "string",
          "policyVersionIdentifier": {
            "policyName": "string",
            "policyVersionId": "string"
          },
          "account": "string"
        },
        "additionalInfo": {
          "string": "string"
        }
      },
      "relatedResources": [
        {
          "resourceType": "string",
          "resourceIdentifier": {
            "deviceCertificateId": "string",
            "caCertificateId": "string",
            "cognitoIdentityPoolId": "string",
            "clientId": "string",

            "iamRoleArn": "string",

            "policyVersionIdentifier": {
              "policyName": "string",
              "policyVersionId": "string"
            },
            "account": "string"
          },

          "roleAliasArn": "string",

          "additionalInfo": {
            "string": "string"
          }
        }
      ],
      "reasonForNonCompliance": "string",
      "reasonForNonComplianceCode": "string"
    }
  ],
  "nextToken": "string"
}
```

Campos de saída da CLI

Name (Nome)	Type	Descrição
descobertas	list membro: AuditFinding	As descobertas (resultados) da auditoria.

Name (Nome)	Type	Descrição
taskId	string comprimento - máx.: 40 mín.: 1 padrão: [a-zA-Z0-9-]+	O ID da auditoria que gerou esse resultado (descoberta).
checkName	string	A verificação de auditoria que gerou esse resultado.
taskStartTime	timestamp	A hora em que a auditoria foi iniciada.
findingTime	timestamp	A hora em que o resultado (descoberta) foi encontrado.
severity	string	A severidade do resultado (descoberta). enum: CRÍTICO | ALTO | MÉDIO | BAIXO
nonCompliantResource	NonCompliantResource	O recurso identificado como incompatível com a verificação de auditoria.
resourceType	string	O tipo do recurso incompatível. enum: DEVICE_CERTIFICATE | CA_CERTIFICATE | IOT_POLICY | COGNITO_IDENTITY_POOL | CLIENT_ID | ACCOUNT_SETTINGS
resourceIdentifier	ResourceIdentifier	As informações que identificam o recurso incompatível.
deviceCertificateId	string comprimento - máx.: 64 mín.: 64 padrão: (0x)?[a-fA-F0-9]+	O ID do certificado anexado ao recurso.
caCertificateId	string comprimento - máx.: 64 mín.: 64 padrão: (0x)?[a-fA-F0-9]+	O ID do certificado CA usado para autorizar o certificado.
cognitoIdentityPoolId	string	O ID do grupo de identidades do Amazon Cognito.
clientId	string	O ID do cliente.
policyVersionIdentifier	PolicyVersionIdentifier	A versão da política associada ao recurso.

Name (Nome)	Type	Descrição
policyName	string comprimento - máx.: 128 mín.: 1 padrão: [w+=,.@-]+	O nome da política.
policyVersionId	string padrão: [0-9] +	O ID da versão da política associada ao recurso.
account	string comprimento - máx.: 12 mín.: 12 padrão: [0-9] +	A conta com a qual o recurso está associado.
additionalInfo	mapear	Outras informações sobre o recurso incompatível.
relatedResources	list membro: RelatedResource	A lista de recursos relacionados.
resourceType	string	O tipo de recurso. enum: DEVICE_CERTIFICATE | CA_CERTIFICATE | IOT_POLICY | COGNITO_IDENTITY_POOL | CLIENT_ID | ACCOUNT_SETTINGS
resourceIdentifier	ResourceIdentifier	As informações que identificam o recurso.
deviceCertificateId	string comprimento - máx.: 64 mín.: 64 padrão: (0x)?[a-fA-F0-9]+	O ID do certificado anexado ao recurso.
caCertificateId	string comprimento - máx.: 64 mín.: 64 padrão: (0x)?[a-fA-F0-9]+	O ID do certificado CA usado para autorizar o certificado.
cognitoIdentityPoolId	string	O ID do grupo de identidades do Amazon Cognito.
clientId	string	O ID do cliente.
policyVersionIdentifier	PolicyVersionIdentifier	A versão da política associada ao recurso.
iamRoleArn	string comprimento - máx.: 2048 mín.: 20	O ARN da função do IAM que tem ações excessivamente permissivas.

Name (Nome)	Type	Descrição
policyName	string comprimento - máx.: 128 mín.: 1 padrão: [w+=,.@-]+	O nome da política.
policyVersionId	string padrão: [0-9] +	O ID da versão da política associada ao recurso.
roleAliasArn	string length- max:2048 min:1	O ARN do alias de função que tem ações excessivamente permissivas.
account	string comprimento - máx.: 12 mín.: 12 padrão: [0-9] +	A conta com a qual o recurso está associado.
additionalInfo	mapear	Outras informações sobre o recurso.
reasonForNonCompliance	string	O motivo pelo qual o recurso era incompatível.
reasonForNonComplianceCode	string	Um código que indica o motivo pelo qual o recurso era incompatível.
nextToken	string	Um token que pode ser usado para recuperar o próximo conjunto de resultados ou `null` se não houver resultados adicionais.

Erros

`InvalidRequestException`

O conteúdo da solicitação era inválido.

`ThrottlingException`

A taxa excede o limite.

`InternalFailureException`

Ocorreu um erro inesperado.

Supressões de descoberta de auditoria

Quando você executa uma auditoria, ela relata descobertas para todos os recursos não compatíveis. Isso significa que seus relatórios de auditoria incluem descobertas de recursos em que você está trabalhando para mitigar problemas e também para recursos que não são compatíveis, como teste ou dispositivos quebrados. A auditoria continua relatando descobertas de recursos que permanecem não compatíveis em execuções de auditoria sucessivas, o que pode adicionar informações indesejadas aos seus relatórios. As

supressões de localização de auditoria permitem suprimir ou filtrar descobertas por um período definido até que o recurso seja corrigido ou indefinidamente para um recurso associado a um teste ou dispositivo quebrado.

> **Note**
>
> Ações de mitigação não estarão disponíveis para descobertas de auditoria suprimidas. Para obter mais informações sobre ações de mitigação, consulteAções de mitigação (p. 1045).

Para obter informações sobre cotas de supressão de descoberta de auditoria, consulteAWS IoTEndpoints e cotas do Device Defender.

Como funcionam as supressões de descoberta de auditoria

Quando você cria uma supressão de descoberta de auditoria para um recurso não compatível, seus relatórios de auditoria e notificações se comportam de maneira diferente.

Seus relatórios de auditoria incluirão uma nova seção que lista todas as descobertas suprimidas associadas ao relatório. As descobertas suprimidas não serão consideradas quando avaliarmos se uma verificação de auditoria está em conformidade ou não. Uma contagem de recursos suprimida também é retornada para cada verificação de auditoria quando você usa odescribe-audit-taskNa interface de linha de comando (CLI).

Para notificações de auditoria, as descobertas suprimidas não são consideradas quando avaliamos se uma verificação de auditoria está em conformidade ou não. Uma contagem de recursos suprimida também está incluída em cada notificação de verificação de auditoriaAWS IoT Device DefenderPublica no Amazon CloudWatch e no Amazon Simple Notification Service (Amazon SNS).

Como usar supressões de descoberta de auditoria no console

Para suprimir uma descoberta de um relatório de auditoria

O procedimento a seguir mostra como criar uma supressão de descoberta de auditoria naAWS IoTconsole do .

1. NoAWS IoTconsoleNo painel de Navegação, expandaDefenderDepois, escolhaAuditoria,Resultados.
2. Selecione um relatório de auditoria que você gostaria de revisar.

3. NoVerificações incompatíveisseção, sobNome da verificação, escolha a verificação de auditoria na qual está interessado.

4. Na tela de detalhes da verificação de auditoria, se houver descobertas que você não deseja ver, selecione o botão de opção ao lado da descoberta. Depois, escolhaAçõese, em seguida, escolha a quantidade de tempo que você gostaria que sua auditoria encontrasse supressão persistisse.

   Note

   No console do, é possível selecionar1 semana,1 mês,3 meses,6 meses, ouIndefinidamentecomo datas de vencimento para a supressão de descoberta de auditoria. Se você quiser definir uma data de expiração específica, poderá fazê-lo somente na CLI ou na API. As supressões de descoberta de auditoria também podem ser canceladas a qualquer momento, independentemente da data de expiração.

5. Confirme os detalhes da supressão e, depois, escolhaHabilitar supressão.

6. Depois de criar a supressão de descoberta de auditoria, um banner aparece confirmando que sua supressão de descoberta de auditoria foi criada.

Para ver descobertas suprimidas em um relatório de auditoria

1. NoAWS IoTconsoleNo painel de Navegação, expandaDefenderDepois, escolhaAuditoria,Resultados.
2. Selecione um relatório de auditoria que você gostaria de revisar.
3. NoDescobertas suprimidas, veja quais descobertas de auditoria foram suprimidas para o relatório de auditoria escolhido.

Para listar suas supressões de descoberta de auditoria

- NoAWS IoTconsoleNo painel de Navegação, expandaDefenderDepois, escolhaAuditoria,Como encontrar supressão.

Para editar a supressão de descoberta de auditoria

1. NoAWS IoTconsoleNo painel de Navegação, expandaDefenderDepois, escolhaAuditoria,Como encontrar supressão.
2. Selecione o botão de opção ao lado da supressão de descoberta de auditoria que você gostaria de editar. Depois, escolhaAções,Edite.
3. NoEditar supressão de descoberta de auditoriajanela, você pode alterar oDuração da supressãoouDescription (opcional).

4. Depois de fazer as alterações, escolhaSalvar. OComo encontrar supressãoJanela é aberta.

Para excluir uma supressão de descoberta de auditoria

1. NoAWS IoTconsoleNo painel de Navegação, expandaDefenderDepois, escolhaAuditoria,Como encontrar supressão.
2. Selecione o botão de opção ao lado da supressão de descoberta de auditoria que você deseja excluir e, depois, escolhaAções,Excluir.
3. NoExcluir supressão de descoberta de auditoriajanela, insira`delete`na caixa de texto para confirmar sua exclusão e, em seguida, escolhaExcluir. OComo encontrar supressãoJanela é aberta.

Como usar supressões de descoberta de auditoria na CLI

Você pode usar os comandos da CLI do para criar e gerenciar supressão de descoberta de auditoria.

- create-audit-supressão
- descrever-supressão de auditoria
- atualização de supressão de auditoria
- supressão de exclusão de auditoria
- supressões de auditoria de lista

O`resource-identifier`você insere depende da`check-name`você está suprimindo descobertas para. A tabela a seguir detalha quais verificações exigem quais`resource-identifier`para criar e editar supressões.

> **Note**
>
> Os comandos de supressão não indicam a desativação de uma auditoria. As auditorias ainda serão executadas no seuAWS IoTDispositivos. As supressões são aplicáveis somente às descobertas da auditoria.

check-name	resource-identifier
AUTHENTICATE_COGNITO_ROLE_OVERLY_PERMISSIVE_CHECK	cognitoIdentityPoolId
CA_CERT_APPROACHING_EXPIRATION_CHECK	caCertificateId
CA_CERTIFICATE_KEY_QUALITY_CHECK	caCertificateId
CONFLICTING_CLIENT_IDS_CHECK	clientId
DEVICE_CERT_APPROACHING_EXPIRATION_CHECK	deviceCertificateId
DEVICE_CERTIFICATE_KEY_QUALITY_CHECK	deviceCertificateId
DEVICE_CERTIFICATE_SHARED_CHECK	deviceCertificateId
IOT_POLICY_OVERLY_PERMISSIVE_CHECK	policyVersionIdentifier
IOT_ROLE_ALIAS_ALLOWS_ACCESS_TO_UNUSED_SERVICES_CHECK	roleAliasARN

check-name	resource-identifier
IOT_ROLE_ALIAS_OVERLY_PERMISSIVE_CHECK	roleAliasArn
LOGGING_DISABLED_CHECK	account
REVOKED_CA_CERT_CHECK	caCertificateId
REVOKED_DEVICE_CERT_CHECK	deviceCertificateId
UNAUTHENTICATED_COGNITO_ROLE_OVERLY_PERMISSIVE_CHECK	cognitoIdentityPoolId

Para criar e aplicar uma supressão de descoberta de auditoria

O procedimento a seguir mostra como criar uma supressão de descoberta de auditoria naAWSCLI.

- Usar `create-audit-suppression`para criar uma supressão de descoberta de auditoria. O exemplo a seguir cria uma supressão de descoberta de auditoria paraConta da AWS *123456789012*com base no chequeRegistro em log desativado.

```
aws iot create-audit-suppression \
    --check-name LOGGING_DISABLED_CHECK \
    --resource-identifier account=123456789012 \
    --client-request-token 28ac32c3-384c-487a-a368-c7bbd481f554 \
    --suppress-indefinitely \
    --description "Suppresses logging disabled check because I don't want to enable
 logging for now."
```

  Não há saída para esse comando.

APIs de supressões de descoberta de auditoria

As APIs a seguir podem ser usadas para criar e gerenciar supressões de localização de auditoria.

- CreateAuditSupressão
- DescribeAuditSupressão
- updateAuditSuppression
- DeleteAuditSupression
- ListAuditSuppressions

Para filtrarpelodescobertas de auditoria específicas, você pode usar oListAuditFindingsAPI.

Detectar

O AWS IoT Device Defender Detect permite identificar comportamento anormal que pode indicar um dispositivo comprometido ao monitorar o comportamento dos dispositivos. Usando uma combinação de métricas da nuvem (da AWS IoT) e métricas do dispositivo (de agentes instalados em seus dispositivos), você pode detectar:

- Mudanças nos padrões de conexão.
- Dispositivos que se comunicam com endpoints não autorizados ou não reconhecidos.
- Alterações nos padrões de tráfego de dispositivos de entrada e saída.

Crie perfis de segurança, que contêm definições de comportamentos esperados do dispositivo, e os atribua a um grupo de dispositivos ou a todos os dispositivos da sua frota.AWS IoT Device Defender O Detect usa esses perfis de segurança para detectar anomalias e enviar alarmes por meio de métricas do Amazon CloudWatch e notificações do Amazon Simple Notification Service.

O AWS IoT Device Defender Detect pode detectar problemas de segurança encontrados frequentemente em dispositivos conectados:

- O tráfego de um dispositivo para um endereço IP mal-intencionado conhecido ou para um endpoint não autorizado que indica um possível comando mal-intencionado e canal de controle.
- Tráfego anormal, como um pico no tráfego de saída, que indica que um dispositivo está participando de um DDoS.
- Dispositivos com interfaces de gerenciamento remoto e portas com acesso remoto.
- Um pico na taxa de mensagens enviadas à sua conta (por exemplo, de um dispositivo invasor que pode resultar em um número excessivo de cobranças por mensagem).

Casos de uso:

Medir a superfície de ataque

Use o AWS IoT Device Defender Detect para avaliar a superfície de ataque dos dispositivos. Por exemplo, você pode identificar dispositivos com portas de serviço que muitas vezes são o alvo de ataques (serviço telnet em execução nas portas 23/2323, serviço SSH em execução na porta 22, serviços HTTP/S em execução nas portas 80/443/8080/8081). Enquanto essas portas de serviço podem ter motivos legítimos para serem usadas nos dispositivos, elas também geralmente fazem parte da superfície de ataque para adversários e carregam os riscos associados. DepoisAWS IoT Device DefenderDepois que o Detect alertar sobre a superfície de ataque, você pode minimizá-la (eliminando os serviços de rede não utilizados) ou executar avaliações adicionais para identificar vulnerabilidades de segurança (por exemplo, telnet configurado com senhas comuns, padrões ou fracas).

Detectar anomalias comportamentais de dispositivos com possíveis causas raiz de segurança

Você pode usarAWS IoT Device DefenderDepois que o Detect alertar sobre métricas comportamentais inesperadas de dispositivos (o número de portas abertas, o número de conexões, uma porta inesperada aberta, conexões com endereços IP inesperados) que podem indicar uma violação da segurança. Por exemplo, um número de conexões TCP maior que o esperado pode indicar que um dispositivo está sendo usado para um ataque DDoS. Uma escuta de processamento em uma porta diferente da esperada pode indicar um backdoor instalado em um dispositivo para controle remoto. Use o AWS IoT Device Defender Detect para verificar a integridade das frotas de dispositivos e verificar os pressupostos de segurança (por exemplo, nenhum dispositivo está em escuta na porta 23 ou 2323).

Você pode habilitar a detecção de ameaças baseada em machine learning (ML) para identificar automaticamente possíveis ameaças.

Detectar um dispositivo configurado incorretamente

Um pico no número ou tamanho de mensagens enviadas de um dispositivo para a sua conta pode indicar um dispositivo configurado incorretamente. Tal dispositivo pode aumentar as cobranças por mensagem. Da mesma forma, um dispositivo com muitas falhas de autorização pode exigir uma política reconfigurada.

Monitorar o comportamento de dispositivos não registrados

O AWS IoT Device Defender Detect permite identificar comportamentos incomuns de dispositivos não registrados na AWS IoT. Você pode definir os perfis de segurança que são específicos para um dos seguintes tipos de destino:

- Todos os dispositivos
- Todos os dispositivos registrados (coisas no registro da AWS IoT)
- Todos os dispositivos não registrados
- Os dispositivos em um grupo de coisas

Um perfil de segurança define um conjunto de comportamentos esperado para dispositivos em sua conta e especifica as ações a serem tomadas quando uma anomalia é detectada. Perfis de segurança devem ser anexados aos destinos mais específicos para oferecer controle granular sobre quais dispositivos estão sendo avaliados com relação a esse perfil.

Os dispositivos não registrados devem fornecer um identificador de cliente ou um nome de coisa MQTT consistente (para dispositivos que relatam métricas de dispositivo) durante a vida útil do dispositivo, de modo que todas as violações e métricas sejam atribuídas ao mesmo dispositivo.

> **Important**
>
> As mensagens relatadas por dispositivos são rejeitadas se o nome da coisa contiver caracteres de controle ou se o nome da coisa tiver mais de 128 bytes de caracteres codificados em UTF-8.

Casos de uso de segurança

Esta seção descreve os diferentes tipos de ataques que ameaçam sua frota de dispositivos e as métricas recomendadas que você pode usar para monitorar esses ataques. Recomendamos usar anomalias métricas como ponto de partida para investigar problemas de segurança, mas você não deve basear sua determinação de ameaças à segurança apenas em uma anomalia métrica.

Para investigar um alarme de anomalia, correlacione os detalhes do alarme com outras informações contextuais, como atributos do dispositivo, tendências históricas de métricas de dispositivo, tendências históricas de métricas de perfil de segurança, métricas personalizadas e logs para determinar se uma ameaça à segurança está presente.

Casos de uso do lado da nuvem

O Device Defender pode monitorar os seguintes casos de uso noAWS IoTdo lado da nuvem.

Roubo de propriedade intelectual:

> O roubo de propriedade intelectual envolve roubar as propriedades intelectuais de uma pessoa ou de empresas, incluindo segredos comerciais, hardware ou software. Muitas vezes ocorre durante o estágio de fabricação dos dispositivos. O roubo de propriedade intelectual pode vir na forma de pirataria, roubo de dispositivos ou roubo de certificados de dispositivo. O roubo de propriedade intelectual baseado em nuvem pode ocorrer devido à presença de políticas que permitem acesso não intencional a recursos de IoT. Você deve revisar seuPolíticas da IoTe Ative oAudite verificações excessivamente permissivaspara identificar políticas excessivamente permissivas.

Métricas relacionadas:

Métrica	Raciocínio
IP de origem	Se o dispositivo for roubado, seu endereço IP de origem ficaria fora do intervalo de endereços IP normalmente esperado para dispositivos circulados em uma cadeia de suprimentos normal.
Número de mensagens recebidas	Como um invasor pode usar um dispositivo em roubo de IP baseado em nuvem, métricas relacionadas a contagens de mensagens ou tamanhos de mensagens enviadas para o dispositivo deAWS IoTa nuvem pode aumentar, indicando um possível problema de segurança.
Tamanho da mensagem	

Exfiltração de dados baseada em MQTT:

A exfiltração de dados ocorre quando um ator mal-intencionado realiza uma transferência de dados não autorizada de uma implantação de IoT ou de um dispositivo. O atacante lança esse tipo de ataques por meio do MQTT contra fontes de dados do lado da nuvem.

Métricas relacionadas:

Métrica	Raciocínio
IP de origem	Se um dispositivo for roubado, seu endereço IP de origem ficaria fora do intervalo de endereços IP normalmente esperado para dispositivos circulados em uma cadeia de suprimentos padrão.
Número de mensagens recebidas	Como um invasor pode usar um dispositivo em uma exfiltração de dados baseada em MQCT, métricas relacionadas a contagens de mensagens ou tamanhos de mensagens enviadas para o dispositivo deAWS IoTa nuvem pode aumentar, indicando um possível problema de segurança.
Tamanho da mensagem	

Personificação:

Um ataque de personificação é onde os invasores se apresentam como entidades conhecidas ou confiáveis em um esforço para acessarAWS IoTserviços, aplicativos, dados no lado da nuvem ou envolvimento no comando e controle de dispositivos IoT.

Métricas relacionadas:

Métrica	Raciocínio
Falhas de autorização	Quando os invasores se apresentam como entidades confiáveis usando identidades roubadas, as métricas relacionadas à conectividade geralmente aumentam, pois as credenciais podem não ser mais válidas
Tentativas do Conneciton	
Desconecta	

Métrica	Raciocínio
	ou podem ser usadas por um dispositivo confiável. Comportamentos anômalos em falhas de autorização, tentativas de conexão ou desconexões apontam para um cenário de representação potencial.

Abuso de infraestrutura em nuvem

AbusoAWS IoTos serviços em nuvem ocorrem ao publicar ou se inscrever em tópicos com um alto volume de mensagens ou com mensagens em tamanhos grandes. Políticas excessivamente permissivas ou exploração de vulnerabilidade de dispositivos para comando e controle também podem causar abuso de infraestrutura na nuvem. Um dos principais objetivos desse ataque é aumentar suaAWSFaturamento. Você deve revisar seuPolíticas da IoTe Ative oAudite verificações excessivamente permissivaspara identificar políticas excessivamente permissivas.

Métricas relacionadas:

Métrica	Raciocínio
Número de mensagens recebidas	O objetivo desse ataque é aumentar suaAWSbill, métricas que monitoram atividades como contagem de mensagens, mensagens recebidas e tamanho da mensagem aumentarão.
Número de mensagens enviadas	
Tamanho da mensagem	
IP de origem	Listas de IP de origem suspeita podem aparecer, a partir das quais os invasores geram seu volume de mensagens.

Casos de uso do lado do dispositivo

O Device Defender pode monitorar os seguintes casos de uso no lado do dispositivo.

Ataque de negação de serviço:

Um ataque de negação de serviço (DoS) visa desligar um dispositivo ou rede, tornando o dispositivo ou a rede inacessível aos usuários pretendidos. Os ataques DoS bloqueiam o acesso inundando o destino com tráfego ou enviando solicitações que iniciam um sistema lento ou fazem com que o sistema falhe. Seus dispositivos IoT podem ser usados em ataques DoS.

Métricas relacionadas:

Métrica	Raciocínio
Pacotes de saída	Os ataques DoS geralmente envolvem taxas mais altas de comunicação de saída de um determinado dispositivo e, dependendo do tipo de ataque DoS, pode haver um aumento em um ou ambos os números de pacotes fora e bytes de saída.
Bytes de saída	
IP de destino	Se você definir os endereços IP/intervalos CIDR com os quais seus dispositivos devem se comunicar, uma anomalia no IP de destino

Métrica	Raciocínio
	poderá indicar comunicação IP não autorizada a partir de seus dispositivos.
Portas TCP de escuta	Um ataque DoS geralmente requer uma infraestrutura maior de comando e controle, onde o malware instalado em seus dispositivos recebe comandos e informações sobre quem atacar e quando atacar. Portanto, para receber essas informações, o malware normalmente escutaria em portas que normalmente não são usadas por seus dispositivos.
Portas TCP de escuta	
Portas UDP de escuta	
Portas UDP de escuta	

Escalonamento lateral de ameaças:

O escalonamento lateral de ameaças geralmente começa com um invasor obtendo acesso a um ponto de uma rede, por exemplo, um dispositivo conectado. Em seguida, o invasor tenta aumentar o nível de privilégios ou o acesso a outros dispositivos por meio de métodos como credenciais roubadas ou explorações de vulnerabilidade.

Métricas relacionadas:

Métrica	Raciocínio
Pacotes de saída	Em situações típicas, o invasor teria que executar uma varredura na rede local para realizar o reconhecimento e identificar os dispositivos disponíveis para restringir sua seleção de alvo de ataque. Esse tipo de varredura pode resultar em um pico de contagens de bytes e pacotes.
Bytes de saída	
IP de destino	Se um dispositivo deve se comunicar com um conjunto conhecido de endereços IP ou CIDRs, você pode identificar se ele tenta se comunicar com um endereço IP anormal, que geralmente seria um endereço IP privado na rede local em um caso de uso de escalonamento de ameaças lateral.
Falhas de autorização	À medida que o invasor tenta aumentar seu nível de privilégios em uma rede IoT, ele pode usar credenciais roubadas que foram revogadas ou expiradas, o que causaria maiores falhas de autorização.

Exfiltração ou vigilância de dados:

A exfiltração de dados ocorre quando um malware ou um ator mal-intencionado realiza uma transferência de dados não autorizada de um dispositivo ou de um endpoint de rede. A exfiltração de dados normalmente serve a dois propósitos para o invasor, obtendo dados ou propriedade intelectual ou realizando reconhecimento de uma rede. Vigilância significa que código malicioso é usado para monitorar as atividades do usuário com a finalidade de roubar credenciais e coletar informações. As métricas abaixo podem fornecer um ponto de partida para investigar qualquer tipo de ataque.

Métricas relacionadas:

Métrica	Raciocínio
Pacotes de saída	Quando ocorrem ataques de exfiltração ou vigilância de dados, o invasor geralmente espelhava os dados enviados do dispositivo em vez de simplesmente redirecionar os dados, o que seria identificado pelo defensor quando não vê os dados pretendidos chegando. Esses dados espelhados aumentariam significativamente a quantidade total de dados enviados do dispositivo, resultando em um pico de pacotes e contagens de bytes.
Bytes de saída	
IP de destino	Quando um invasor está usando um dispositivo em ataques de exfiltração de dados ou vigilância, os dados precisariam ser enviados para um endereço IP anormal controlado pelo invasor. Monitorar o IP de destino pode ajudar a identificar esse ataque.

Mineração de criptomoedas

Os invasores aproveitam o poder de processamento de dispositivos para minerar criptomoedas. A mineração de criptografia é um processo computacionalmente intensivo, normalmente exigindo comunicação de rede com outros pares e pools de mineração.

Métricas relacionadas:

Métrica	Raciocínio
IP de destino	A comunicação de rede geralmente é um requisito durante a criptografia. Ter uma lista rigorosamente controlada de endereços IP com os quais o dispositivo deve se comunicar pode ajudar a identificar a comunicação não intencional em um dispositivo, como a mineração de criptomoedas.
Uso da CPUMétrica personalizada do	A mineração de criptomoedas requer computação intensiva, resultando em alta utilização da CPU do dispositivo. Se você optar por coletar e monitorar essa métrica, um uso de CPU maior do que o normal pode ser um indicador de atividades de mineração de criptografia.

Comando e controle, malware e ransomware

Malware ou ransomware restringe seu controle sobre seus dispositivos e limita a funcionalidade do dispositivo. No caso de um ataque de ransomware, o acesso aos dados seria perdido devido à criptografia que o ransomware usa.

Métricas relacionadas:

Métrica	Raciocínio
IP de destino	Ataques de rede ou remotos representam uma grande parte dos ataques em dispositivos IoT. Uma lista bem controlada de endereços IP com os quais o dispositivo deve se comunicar pode ajudar a identificar IPs de destino anormais resultantes de um ataque de malware ou ransomware.
Portas TCP de escuta	Vários ataques de malware envolvem iniciar um servidor de comando e controle que envia comandos para execução em um dispositivo. Esse tipo de servidor é essencial para uma operação de malware ou ransomware e pode ser identificado monitorando rigorosamente as portas TCP/UDP abertas e as contagens de portas.
Portas TCP de escuta	
Portas UDP de escuta	
Portas UDP de escuta	

Conceitos

métrica

AWS IoT Device DefenderO Detect usa métricas para detectar comportamentos anormais dos dispositivos.AWS IoT Device Defender O Detect compara o valor reportado de uma métrica com o valor esperado fornecido. Essas métricas podem ser obtidas de duas fontes: métricas no lado da nuvem e métricas no lado do dispositivo. O ML Detect suporta 6 métricas no lado da nuvem e 7 métricas no lado do dispositivo. Para ver uma lista das métricas com suporte para Detect de ML, consulteMétricas compatíveis (p. 1005).

O comportamento anormal na rede da AWS IoT é detectado usando métricas no lado da nuvem, como o número de falhas de autorização, ou o número ou tamanho de mensagens que um dispositivo envia ou recebe por meio da AWS IoT.

AWS IoT Device DefenderO Detect também pode coletar, agregar e monitorar dados de métricas gerados porAWS IoTDispositivos (por exemplo, as portas em que um dispositivo está em escuta, o número de bytes ou pacotes enviados ou as conexões TCP do dispositivo).

Use o AWS IoT Device Defender Detect com métricas no lado da nuvem isoladamente. Para usar métricas no lado do dispositivo, primeiro é necessário implantar o SDK da AWS IoT nos dispositivos conectados à AWS IoT ou gateways de dispositivo para coletar as métricas e enviá-las à AWS IoT. Consulte Envio de métricas de dispositivos (p. 1027).

Perfil de segurança

Um perfil de segurança define comportamentos anômalos para um grupo de dispositivos (umGrupo de coisas (p. 272)) ou para todos os dispositivos em sua conta e especifica quais ações tomar quando uma anomalia for detectada. Você pode usar oAWS IoTComandos de console ou API para criar um Perfil de segurança e associá-lo a um grupo de dispositivos.AWS IoT Device Defender O Detect começa a registrar dados relacionados à segurança e usa os comportamentos definidos no Perfil de segurança para detectar anomalias no comportamento dos dispositivos.

comportamento

Um comportamento informaAWS IoT Device DefenderDetect como reconhecer quando um dispositivo está fazendo algo anormal. Qualquer ação de dispositivo que não corresponder a um comportamento acionará um alerta. Um comportamento de Detecção de Regras consiste em uma métrica e um valor

absoluto ou limite estatístico com um operador (por exemplo, menor ou igual a, maior ou igual a), que descrevem o comportamento esperado do dispositivo. Um comportamento de Detecção de ML consiste em uma métrica e uma configuração de Detecção de ML, que definem um modelo de ML para aprender o comportamento normal dos dispositivos.

modelo de ML

Um modelo de ML é um modelo de aprendizado de máquina criado para monitorar cada comportamento que um cliente configura. O modelo treina em padrões de dados métricos de grupos de dispositivos direcionados e gera três limites de confiança de anomalias (alto, médio e baixo) para o comportamento baseado em métrica. Ele infere anomalias com base em dados métricos ingeridos no nível do dispositivo. No contexto do ML Detect, um modelo de ML é criado para avaliar um comportamento baseado em métrica. Para obter mais informações, consulte Detecção de ML (p. 1003).

Nível de confiança

O ML Detect suporta três níveis de confiança:`High`,`Medium`, e`Low`.`High`confiança significa baixa sensibilidade na avaliação do comportamento anômalo e frequentemente um menor número de alarmes.`Medium`confiança significa sensibilidade média e`Low`confiança significa alta sensibilidade e frequentemente um maior número de alarmes.

dimensão

Você pode definir uma dimensão para ajustar o escopo de um comportamento. Por exemplo, você pode definir uma dimensão de filtro de tópico que aplica um comportamento a tópicos MQTT que correspondem a um padrão. Para obter informações sobre como definir uma dimensão para uso em um Perfil de segurança, consulteCreateDimension.

alarme

Quando uma anomalia for detectada, uma notificação de alarme pode ser enviada através de um CloudWatch métrica (consulteUsar métricas do AWS IoT (p. 439)) ou uma notificação do SNS. Uma notificação de alarme também é exibida naAWS IoTConsole junto com informações sobre o alarme e um histórico de alarmes para o dispositivo. Um alarme também será enviado quando um dispositivo monitorado parar de exibir comportamento anormal ou quando estava causando um alarme mas para de relatar durante um período estendido.

Verification state de alarme

Depois que um alarme for criado, você pode verificar o alarme como Verdadeiro positivo, Benigno positivo, Falso positivo ou Desconhecido. Você também pode adicionar uma descrição ao estado de verificação de alarme. Você pode visualizar, organizar e filtrarAWS IoT Device Defenderalarmes usando um dos quatro estados de verificação. Você pode usar estados de verificação de alarmes e descrições relacionadas para informar os membros da sua equipe. Isso ajuda sua equipe a tomar ações de acompanhamento, por exemplo, realizar ações de mitigação em alarmes positivos verdadeiros, ignorar alarmes positivos benignos ou continuar a investigação sobre alarmes Desconhecidos. O estado de verificação padrão para todos os alarmes é Desconhecido.

Supressão de alarme

Gerenciar Detectar notificações SNS de alarme configurando a notificação de comportamento para`on`o`u`s`u`p`p`r`e`s`s`e`d`. A supressão de alarmes não impede Detectar de realizar avaliações de comportamento do dispositivo; Detect continua sinalizando comportamentos anômalos como alarmes de violação. No entanto, alarmes suprimidos não seriam encaminhados para notificação do SNS. Eles só podem ser acessados através doAWS IoTconsole ou API.

Comportamentos

Um perfil de segurança contém um conjunto de comportamentos. Cada comportamento contém uma métrica que especifica o comportamento normal de um grupo de dispositivos ou de todos os dispositivos

da conta. Os comportamentos se encaixam em duas categorias: Regras Detecte comportamentos e comportamentos de detecção de ML. Com os comportamentos de detecção de regras, você define como seus dispositivos devem se comportar, enquanto o ML Detect usa modelos de ML criados com base em dados históricos de dispositivos para avaliar como seus dispositivos devem se comportar.

Um Perfil de Segurança pode ser de dois tipos de limite: MLouCom base em Regras. Os perfis de segurança da ML detectam automaticamente anomalias operacionais e de segurança no nível do dispositivo em toda a sua frota, aprendendo com dados anteriores. Os perfis de segurança baseados em regras exigem que você defina manualmente regras estáticas para monitorar os comportamentos do dispositivo.

Veja a seguir a descrição de alguns dos campos que são usados na definição de um `behavior`:

Comum à detecção de regras e detecção de ML

`name`

O nome do comportamento.

`metric`

O nome da métrica usada (ou seja, o que é medido pelo comportamento).

`consecutiveDatapointsToAlarm`

Ocorrerá um alarme caso um dispositivo esteja violando o comportamento do número especificado de pontos de dados consecutivos. Se não especificado, o padrão será 1.

`consecutiveDatapointsToClear`

Se um alarme tiver ocorrido e o dispositivo infrator não estiver mais violando o comportamento do número especificado de pontos de dados consecutivos, o alarme será inativado. Se não especificado, o padrão será 1.

`threshold type`

Um Perfil de Segurança pode ser de dois tipos de limite: Baseado em ML ou Regras. Os perfis de segurança da ML detectam automaticamente anomalias operacionais e de segurança no nível do dispositivo em toda a sua frota, aprendendo com dados anteriores. Os perfis de segurança baseados em regras exigem que você defina manualmente regras estáticas para monitorar os comportamentos do dispositivo.

`alarm suppressions`

Gerenciar Detectar notificações SNS de alarme configurando a notificação de comportamento para`onousuppressed`. A supressão de alarmes não impede Detectar de realizar avaliações de comportamento do dispositivo; Detect continua sinalizando comportamentos anômalos como alarmes de violação. No entanto, alarmes suprimidos não são encaminhados para notificação do SNS. Eles só podem ser acessados através doAWS IoTconsole ou API.

Regras Detecção

`dimension`

Você pode definir uma dimensão para ajustar o escopo de um comportamento. Por exemplo, você pode definir uma dimensão de filtro de tópico que aplica um comportamento a tópicos MQTT que correspondem a um padrão. Para definir uma dimensão para uso em um Perfil de Segurança, consulteCreateDimension. Aplica-se somente a Regras Detectar.

`criteria`

Os critérios que determinam se um dispositivo está se comportando normalmente em relação a `metric`.

`comparisonOperator`

O operador que relaciona a coisa medida (`metric`) aos critérios (`value` ou `statisticalThreshold`).

Os possíveis valores são: "less-than", "less-than-equals", "greater-than", "greater-than-equals", "in-cidr-set", "not-in-cidr-set", "in-port-set" e "not-in-port-set". Nem todos os operadores são válidos para todas as métricas. Operadores para portas e conjuntos CIDR são apenas para uso com métricas envolvendo essas entidades.

`value`

O valor a ser comparado com o `metric`. Dependendo do tipo de métrica, este deve conter um `count` (um valor), `cidrs` (uma lista de CIDRs) ou `ports` (uma lista de portas).

`statisticalThreshold`

O limite estatístico pelo qual uma violação de comportamento é determinada. Esse campo contém um campo `statistic` que possui os seguintes valores possíveis: "p0", "p0.1", "p0.01", "p1", "p10", "p50", "p90", "p99", "p99.9", "p99.99" ou "p100".

Essa `statistic` indica um percentil. O resultado é um valor pelo qual a conformidade com o comportamento é determinada. As métricas são coletadas uma ou mais vezes ao longo da duração especificada (`durationSeconds`) de todos os dispositivos de relatório associados a esse Perfil de Segurança, e os percentis são calculados com base nesses dados. Depois disso, as medições são coletadas para um dispositivo e acumuladas ao longo da mesma duração. Se o valor resultante para o dispositivo ficar acima ou abaixo (`comparisonOperator`) do valor associado ao percentil especificado, o dispositivo será considerado em conformidade com o comportamento. Caso contrário, o dispositivo está em violação do comportamento.

Um percentil indica a porcentagem de todas as medidas consideradas abaixo do valor associado. Por exemplo, se o valor associado a "p90" (90º percentil) for 123, 90% de todas as medidas ficaram abaixo de 123.

`durationSeconds`

Use esta opção para especificar o período durante o qual o comportamento é avaliado, para esses critérios que têm uma dimensão de tempo (por exemplo, `NUM_MESSAGES_SENT`). Para uma comparação métrica `statisticalThreshhold`, esse é o período durante o qual as medições são coletadas para todos os dispositivos para determinar os valores `statisticalThreshold` e depois para cada dispositivo para determinar como seu comportamento é classificado em comparação.

Detecção de ML

`ML Detect confidence`

O ML Detect suporta três níveis de confiança:`High`,`Medium`, e`Low`.`High`confiança significa baixa sensibilidade na avaliação do comportamento anômalo e frequentemente um menor número de alarmes,`Medium`confiança significa sensibilidade média, e`Low`confiança significa alta sensibilidade e frequentemente um maior número de alarmes.

Detecção de ML

Com o Machine Learning Detect (ML Detect), você cria perfis de segurança que usam aprendizado de máquina para aprender comportamentos de dispositivos esperados criando automaticamente modelos com base em dados históricos de dispositivos e atribui esses perfis a um grupo de dispositivos ou a todos os dispositivos em sua frota.AWS IoT Device Defenderem seguida, identifica anomalias e aciona alarmes usando os modelos ML.

Para obter informações sobre como começar a usar o ML Detect, consulteGuia de detecção de ML (p. 896).

Este capítulo contém as seguintes seções:

- Casos de uso de ML Detect (p. 1004)
- Como o ML Detect funciona (p. 1004)
- Requisitos mínimos (p. 1004)
- Limitações (p. 1005)
- Marcação de falsos positivos e outros estados de verificação em alarmes (p. 1005)
- Métricas compatíveis (p. 1005)
- Service Quotas (p. 1006)
- Comandos de ML Detect CLI do (p. 1006)
- APIs de detecção de ML (p. 1006)
- Pausar ou excluir um perfil de segurança de detecção de ML (p. 1007)

Casos de uso de ML Detect

Você pode usar o ML Detect para monitorar os dispositivos da frota quando for difícil definir os comportamentos esperados dos dispositivos. Por exemplo, para monitorar o número de métricas de desconexões, talvez não fique claro o que é considerado um limite aceitável. Nesse caso, você pode habilitar o ML Detect para identificar pontos de dados métricos de desconexão anômalos com base em dados históricos relatados a partir de dispositivos.

Outro caso de uso do ML Detect é monitorar comportamentos de dispositivos que mudam dinamicamente ao longo do tempo. O ML Detect aprende periodicamente os comportamentos dinâmicos esperados do dispositivo com base na alteração dos padrões de dados dos dispositivos. Por exemplo, o volume enviado da mensagem do dispositivo pode variar entre dias de semana e fins de semana, e a detecção de ML aprenderá esse comportamento dinâmico.

Como o ML Detect funciona

Usando o ML Detect, você pode criar comportamentos para identificar anomalias operacionais e de segurança em6 métricas do lado da nuvem (p. 1005)e7 métricas do lado do dispositivo (p. 1005). Após o período inicial de treinamento do modelo, o ML Detect atualiza os modelos diariamente com base nos 14 dias finais de dados. Ele monitora os pontos de dados dessas métricas com os modelos de ML e aciona um alarme se uma anomalia for detectada.

O ML Detect funciona melhor se você anexar um perfil de segurança a uma coleção de dispositivos com comportamentos esperados semelhantes. Por exemplo, se alguns de seus dispositivos forem usados nas casas dos clientes e outros dispositivos em escritórios de negócios, os padrões de comportamento do dispositivo podem diferir significativamente entre os dois grupos. Você pode organizar os dispositivos em umdispositivo domésticogrupo de coisas e umdispositivo de escritóriogrupo de coisas. Para obter a melhor eficácia de detecção de anomalias, conecte cada grupo de coisas a um Perfil de Segurança de Detecção de ML separado.

Enquanto o ML Detect está construindo o modelo inicial, ele requer 14 dias e um mínimo de 25.000 pontos de dados por métrica durante o período final de 14 dias para gerar um modelo. Depois, ele atualiza o modelo todos os dias, há um número mínimo de pontos de dados métricos. Se o requisito mínimo não for atendido, o ML Detect tentará criar o modelo no dia seguinte e tentará novamente diariamente pelos próximos 30 dias antes de interromper o modelo para avaliações.

Requisitos mínimos

Para treinar e criar o modelo inicial de ML, o ML Detect tem os seguintes requisitos mínimos.

Período de treinamento mínimo

Leva 14 dias para que os modelos iniciais sejam construídos. Depois disso, o modelo é atualizado todos os dias com dados métricos de um período final de 14 dias.

Pontos de dados totais mínimos

Os datapoints mínimos necessários para criar um modelo de ML são de 25.000 datapoints por métrica nos últimos 14 dias. Para treinamento contínuo e atualização do modelo, o ML Detect requer que os pontos de dados mínimos sejam atendidos a partir de dispositivos monitorados. É aproximadamente o equivalente às seguintes configurações:

- 60 dispositivos conectando e tendo atividadeAWS IoTEm intervalos de 45 minutos.
- 40 dispositivos em intervalos de 30 minutos.
- 15 dispositivos em intervalos de 10 minutos.
- 7 dispositivos em intervalos de 5 minutos.

Alvos de grupos de dispositivos

Para que a coleta de dados progrida, você deve ter coisas nos grupos de coisas de destino para o Perfil de segurança.

Depois que o modelo inicial é criado, os modelos de ML são atualizados todos os dias e exigem pelo menos 25.000 datapoints para o período final de 14 dias.

Limitações

No momento, você não pode usar ML Detect com dimensões ou com métricas personalizadas. As seguintes métricas não são suportadas com ML Detect.

Métricas do lado da nuvem não são suportadas com ML Detect:

- IP de origem (aws:source-ip-address) (p. 1032)

Métricas do lado do dispositivo não são suportadas com ML Detect:

- IPs de destino (`aws:destination-ip-addresses`) (p. 1020)
- Portas TCP de escuta (`aws:listening-tcp-ports`) (p. 1020)
- Portas UDP de escuta (`aws:listening-udp-ports`) (p. 1021)

Marcação de falsos positivos e outros estados de verificação em alarmes

Se você verificar se um alarme de Detecção de ML é um falso positivo por meio de sua investigação, você pode definir o estado de verificação do alarme como Falso positivo. Isso pode ajudar você e sua equipe a identificar alarmes aos quais você não precisa responder. Você também pode marcar alarmes como Verdadeiro positivo, Benigno positivo ou Desconhecido.

Você pode marcar alarmes através doAWS IoT Device Defenderconsoleou usando oPutVerificationStateOnViolationAção de API.

Métricas compatíveis

Você pode usar as seguintes métricas do lado da nuvem com ML Detect:

- Falhas de autorização (aws:num-authorization-failures) (p. 1031)
- Tentativas de conexão (aws:num-connection-tentativas) (p. 1033)
- Desconecta (aws:num-disconnect) (p. 1034)

- Tamanho da mensagem (aws:message-byte-size) (p. 1028)
- Mensagens enviadas (aws:num-messages-sent) (p. 1029)
- Mensagens recebidas (aws:num-messages-received) (p. 1030)

Você pode usar as seguintes métricas do lado do dispositivo com ML Detect:

- Saída de bytes (`aws:all-bytes-out`) (p. 1013)
- Bytes em (`aws:all-bytes-in`) (p. 1014)
- Contagem de portas TCP escutando (`aws:num-listening-tcp-ports`) (p. 1015)
- Contagem de portas UDP escutando (`aws:num-listening-udp-ports`) (p. 1016)
- Pacotes de saída (`aws:all-packets-out`) (p. 1017)
- Pacotes em (`aws:all-packets-in`) (p. 1018)
- Contagem de conexões TCP estabelecidas (`aws:num-established-tcp-connections`) (p. 1021)

Service Quotas

Para obter informações sobre cotas e limites de serviço de ML Detect, consulteAWS IoT Device DefenderEndpoints e cotas do.

Comandos de ML Detect CLI do

Você pode usar os comandos da CLI do para criar e gerenciar ML Detect.

- create-security-profile
- attach-security-profile
- list-security-profiles
- describe-security-profile
- update-security-profile
- delete-security-profile
- resumos de treinamento-modelo de get-behavior-model-training
- list-ative-violações
- list-violação-eventos

APIs de detecção de ML

As APIs a seguir podem ser usadas para criar e gerenciar perfis de segurança de detecção de ML.

- CreateSecurityProfile
- AttachSecurityProfile
- ListSecurityProfiles
- DescribeSecurityProfile
- UpdateSecurityProfile
- DeleteSecurityProfile
- GetBehaviorModelTrainingSummaries
- ListActiveViolations
- ListViolationEvents
- PutVerificationStateOnViolation

Pausar ou excluir um perfil de segurança de detecção de ML

Você pode pausar o Perfil de Segurança de Detecção de ML para interromper o monitoramento temporário dos comportamentos do dispositivo ou excluir o Perfil de Segurança de Detecção de ML para interromper o monitoramento dos comportamentos do dispositivo por um longo período de tempo.

Pausar o perfil de segurança do ML Detect usando o console

Para pausar um Perfil de Segurança de Detecção de ML usando o console, primeiro você deve ter um grupo de coisas vazio. Para criar um grupo de coisas vazio, consulteGrupos de coisas estáticas (p. 272). Se você criou um grupo de coisas vazio, defina o grupo de coisas vazio como o destino do ML Detect Security Profile.

> **Note**
>
> Você precisa definir o destino do seu Perfil de segurança de volta para um grupo de dispositivos com dispositivos dentro de 30 dias ou não poderá reativar o Perfil de segurança.

Exclua ML Detect Security Profile usando o console

Para excluir um perfil de segurança, siga estas etapas:

1. NoAWS IoTconsole navegue até a barra lateral e escolha oDefenderseção.
2. UnderDefender, escolhaDetectare depoisPerfis de segurança.
3. Escolha o ML Detect Security Profile que você deseja excluir.
4. SelecioneAçõese, em seguida, a partir das opções, escolhaExcluir.

> **Note**
>
> Depois que um Perfil de Segurança de Detecção de ML for excluído, você não poderá reativar o Perfil de Segurança.

Pausar um perfil de segurança de detecção de ML usando a CLI

Para pausar um Perfil de Segurança de Detecção de ML usando a CLI, use `odetach-security-security-profile`Comando da :

```
$aws iot detach-security-profile --security-profile-name SecurityProfileName --
security-profile-target-arn arn:aws:iot:us-east-1:123456789012:all/registered-things
```

> **Note**
>
> Essa opção só está disponível noAWSCLI. Semelhante ao fluxo de trabalho do console, você precisa definir o destino do seu Perfil de Segurança de volta para um grupo de dispositivos com dispositivos dentro de 30 dias, ou não poderá reativar o Perfil de Segurança. Para anexar um perfil de segurança a um grupo de dispositivos, use o`attach-security-profile`comando.

Exclua um perfil de segurança de detecção de ML usando a CLI

Você pode excluir um perfil de segurança usando o`delete-security-profile`comando abaixo:

```
delete-security-profile --security-profile-name SecurityProfileName
```

> **Note**
>
> Depois que um Perfil de Segurança de Detecção de ML for excluído, você não poderá reativar o Perfil de Segurança.

Métricas personalizadas do

comAWS IoT Device Defendermétricas personalizadas, você pode definir e monitorar métricas exclusivas para sua frota ou caso de uso, como número de dispositivos conectados a gateways Wi-Fi, níveis de carga para baterias ou número de ciclos de energia para plugues inteligentes. Os comportamentos de métrica personalizados são definidos nos Perfis de segurança, que especificam comportamentos esperados para um grupo de dispositivos (um grupo de coisas) ou para todos os dispositivos. Você pode monitorar comportamentos configurando alarmes, que podem ser usados para detectar e responder a problemas específicos dos dispositivos.

Este capítulo contém as seguintes seções:

- Como usar métricas personalizadas no console (p. 1008)
- Como usar métricas personalizadas da CLI (p. 1010)
- Comandos personalizados da CLI de métricas personalizadas (p. 1012)
- APIs personalizadas (p. 1013)

Como usar métricas personalizadas no console

Tutoriais

- AWS IoT Device DefenderSDK do agente (Python) (p. 1008)
- Crie uma métrica personalizada e adicione-a a um perfil de segurança (p. 1008)
- Visualizar detalhes personalizados (p. 1009)
- Atualizar uma métrica personalizada (p. 1009)
- Excluir uma métrica personalizada (p. 1009)

AWS IoT Device DefenderSDK do agente (Python)

Para começar, faça download doAWS IoT Device DefenderAgente de amostra do SDK do agente (Python). O agente reúne as métricas e publica relatórios. Depois que as métricas do lado do dispositivo estiverem publicando, você poderá visualizar as métricas que estão sendo coletadas e determinar limites para configurar alarmes. As instruções para configurar o agente do dispositivo estão disponíveis noAWS IoTReadme do SDK do Device Defender Agent (Python). Para obter mais informações, consulteAWS IoT Device DefenderSDK do agente (Python).

Crie uma métrica personalizada e adicione-a a um perfil de segurança

O procedimento a seguir mostra como criar uma métrica personalizada no console do.

1. NoAWS IoTconsole, no painel de navegação, expandaDefendere depois escolhaDetectar,Métricas.
2. NoMétricas personalizadasPage (página), escolhaCriar.
3. NoCriar métrica personalizadaPage (página), faça o seguinte:
    1. UNDERName (Nome), insira um nome para a métrica personalizada. Você não pode modificar esse nome depois de criar a métrica personalizada.
    2. UNDERNome de exibição (opcional)Você pode inserir um nome amigável para sua métrica personalizada. Ele não precisa ser único e pode ser modificado após a criação.
    3. UNDERTipo, escolha o tipo de métrica que você gostaria de monitorar. Os tipos de métricas incluemString-list,lista de endereços ip,lista de números, enúmero. O tipo não pode ser modificado após a criação.
    4. UNDERTagsVocê pode selecionar as tags para serem associadas ao recurso.

Quando tiver terminado, escolhaConfirmar.

4. Depois de criar sua métrica personalizada, oMétricas personalizadasÉ exibida uma página na qual é possível ver sua métrica personalizada recém-criada.
5. Em seguida, é necessário adicionar sua métrica personalizada a um perfil de segurança. NoAWS IoTconsole, no painel de navegação, expandaDefendere depois escolhaDetectar,Perfis de segurança.
6. Escolha o perfil de segurança ao qual você gostaria de adicionar sua métrica personalizada.
7. Selecione Ações, Editar.
8. SelecioneMétricas adicionais para retere, em seguida, escolha sua métrica personalizada. SelecionePróximonas telas a seguir até chegar aoConfirmar. SelecioneSalvareContinuar. Depois que sua métrica personalizada tiver sido adicionada com sucesso, a página de detalhes do Perfil de segurança será exibida.

   Note

   As estatísticas de percentil não estão disponíveis para métricas quando qualquer um dos valores de métrica são números negativos.

Visualizar detalhes personalizados

O procedimento a seguir mostra como visualizar os detalhes de uma métrica personalizada no console do.

1. NoAWS IoTconsole, no painel de navegação, expandaDefendere depois escolhaDetectar,Métricas.
2. Selecione oNome da métricada métrica personalizada da qual você gostaria de ver os detalhes.

Atualizar uma métrica personalizada

O procedimento a seguir mostra como atualizar uma métrica personalizada no console do.

1. NoAWS IoTconsole, no painel de navegação, expandaDefendere depois escolhaDetectar,Métricas.
2. Escolha o botão de opção ao lado da métrica personalizada que você deseja atualizar. Então, paraAções, escolhaEdite.
3. NoAtualizar métrica personalizada, você pode editar o nome de exibição e remover ou adicionar tags.
4. Depois de concluir, escolhaAtualização. OMétricas personalizadas.

Excluir uma métrica personalizada

O procedimento a seguir mostra como excluir uma métrica personalizada no console do.

1. Primeiro, remova sua métrica personalizada de qualquer Perfil de segurança em que ela seja referenciada. Você pode ver quais perfis de segurança contêm sua métrica personalizada na página de detalhes da métrica personalizada. NoAWS IoTconsole, no painel de navegação, expandaDefendere depois escolhaDetectar,Métricas.
2. Escolha a métrica personalizada que você gostaria de remover. Remova a métrica personalizada de qualquer Perfil de segurança listado emPerfis de segurançana página de detalhes da métrica personalizada.
3. NoAWS IoTconsole, no painel de navegação, expandaDefendere depois escolhaDetectar,Métricas.
4. Escolha o botão de opção ao lado da métrica personalizada que você deseja excluir. Então, paraAções, escolhaExcluir.
5. NoTem certeza de que deseja excluir a métrica personalizada?Message, escolhaExcluir métrica personalizada.

Warning

Depois de excluir uma métrica personalizada, você perde todos os dados associados à métrica. Essa ação não pode ser desfeita.

Como usar métricas personalizadas da CLI

Tutoriais

- AWS IoT Device DefenderSDK do agente (Python) (p. 1010)
- Crie uma métrica personalizada e adicione-a a um perfil de segurança (p. 1010)
- Visualizar detalhes personalizados (p. 1011)
- Atualizar uma métrica personalizada (p. 1011)
- Excluir uma métrica personalizada (p. 1012)

AWS IoT Device DefenderSDK do agente (Python)

Para começar, faça download doAWS IoT Device DefenderAgente de amostra do SDK do agente (Python). O agente reúne as métricas e publica relatórios. Depois que as métricas do lado do dispositivo estiverem publicando, você poderá visualizar as métricas que estão sendo coletadas e determinar limites para configurar alarmes. As instruções para configurar o agente do dispositivo estão disponíveis noAWS IoTReadme do SDK do Device Defender Agent (Python). Para obter mais informações, consulteAWS IoT Device DefenderSDK do agente (Python).

Crie uma métrica personalizada e adicione-a a um perfil de segurança

O procedimento a seguir mostra como criar uma métrica personalizada e adicioná-la a um perfil de segurança da CLI.

1. Usar a`create-custom-metric`para criar sua métrica personalizada. O exemplo a seguir cria uma métrica personalizada que mede a porcentagem da bateria.

```
aws iot create-custom-metric \
    --metric-name "batteryPercentage" \
    --metric-type "number" \
    --display-name "Remaining battery percentage." \
    --region us-east-1
    --client-request-token "02ccb92b-33e8-4dfa-a0c1-35b181ed26b0" \
```

   Resultado:

```
{
    "metricName": "batteryPercentage",
    "metricArn": "arn:aws:iot:us-east-1:1234564789012:custommetric/batteryPercentage"
}
```

2. Depois de criar sua métrica personalizada, você pode adicionar a métrica personalizada a um perfil existente usando`update-security-profile`Ou crie um novo perfil de segurança para adicionar a métrica personalizada ao uso`create-security-profile`. Aqui, criamos um novo perfil de segurança chamado*`Uso da bateria`*para adicionar nosso novo*`Porcentagem de bateria`*Métrica personalizada para. Também adicionamos uma métrica de detecção de regras chamada*`Largura de banda celular`*.

```
aws iot create-security-profile \
    --security-profile-name batteryUsage \
```

```
    --security-profile-description "Shows how much battery is left in percentile."  \
    --behaviors "[{\"name\":\"great-than-75\",\"metric\":\"batteryPercentage\",
\"criteria\":{\"comparisonOperator\":\"greater-than\",\"value\":{\"number
\":75},\"consecutiveDatapointsToAlarm\":5,\"consecutiveDatapointsToClear
\":1}},{\"name\":\"cellularBandwidth\",\"metric\":\"aws:message-byte-size\",
\"criteria\":{\"comparisonOperator\":\"less-than\",\"value\":{\"count\":128},
\"consecutiveDatapointsToAlarm\":1,\"consecutiveDatapointsToClear\":1}}]" \
    --region us-east-1
```

Resultado:

```
{
    "securityProfileArn": "arn:aws:iot:us-east-1:1234564789012:securityprofile/
batteryUsage",
    "securityProfileName": "batteryUsage"
}
```

Note

As estatísticas de percentil não estão disponíveis para métricas quando qualquer um dos valores de métrica são números negativos.

Visualizar detalhes personalizados

O procedimento a seguir mostra como visualizar os detalhes de uma métrica personalizada da CLI.

- Usar a`list-custom-metrics`comando para visualizar todas as métricas personalizadas.

```
aws iot list-custom-metrics \
    --region us-east-1
```

A saída desse comando é semelhante à seguinte.

```
{
    "metricNames": [
        "batteryPercentage"
    ]
}
```

Atualizar uma métrica personalizada

O procedimento a seguir mostra como atualizar uma métrica personalizada da CLI.

- Usar a`update-custom-metric`Comando para atualizar uma métrica personalizada. O exemplo a seguir atualiza o`display-name`.

```
aws iot update-custom-metric \
    --metric-name batteryPercentage \
    --display-name 'remaining battery percentage on device' \
    --region us-east-1
```

A saída desse comando é semelhante à seguinte.

```
{
    "metricName": "batteryPercentage",
    "metricArn": "arn:aws:iot:us-east-1:1234564789012:custommetric/batteryPercentage",
```

```
    "metricType": "number",
    "displayName": "remaining battery percentage on device",
    "creationDate": "2020-11-17T23:01:35.110000-08:00",
    "lastModifiedDate": "2020-11-17T23:02:12.879000-08:00"
}
```

Excluir uma métrica personalizada

O procedimento a seguir mostra como excluir uma métrica personalizada da CLI.

1. Para excluir uma métrica personalizada, primeiro remova-a de todos os perfis de segurança aos quais ela está anexada. Usar a`list-security-profiles`para exibir Perfis de segurança com uma determinada métrica personalizada.
2. Para remover uma métrica personalizada de um Perfil de segurança, use o`update-security-profiles`comando. Insira todas as informações que deseja manter, mas exclua a métrica personalizada.

```
aws iot update-security-profile \
  --security-profile-name batteryUsage \
  --behaviors "[{\"name\":\"cellularBandwidth\",\"metric\":\"aws:message-byte-size
\",\"criteria\":{\"comparisonOperator\":\"less-than\",\"value\":{\"count\":128},
\"consecutiveDatapointsToAlarm\":1,\"consecutiveDatapointsToClear\":1}}]"
```

   A saída desse comando é semelhante à seguinte.

```
{
  "behaviors": [{\"name\":\"cellularBandwidth\",\"metric\":\"aws:message-byte-size
\",\"criteria\":{\"comparisonOperator\":\"less-than\",\"value\":{\"count\":128},
\"consecutiveDatapointsToAlarm\":1,\"consecutiveDatapointsToClear\":1}}],
  "securityProfileName": "batteryUsage",
  "lastModifiedDate": 2020-11-17T23:02:12.879000-09:00,
  "securityProfileDescription": "Shows how much battery is left in percentile.",
  "version": 2,
  "securityProfileArn": "arn:aws:iot:us-east-1:1234564789012:securityprofile/
batteryUsage",
  "creationDate": 2020-11-17T23:02:12.879000-09:00
}
```

3. Depois que a métrica personalizada for desanexada, use a`delete-custom-metric`Comando para excluir a métrica personalizada.

```
aws iot delete-custom-metric  \
  --metric-name batteryPercentage \
  --region us-east-1
```

   A saída desse comando é semelhante à seguinte

```
HTTP 200
```

Comandos personalizados da CLI de métricas personalizadas

Você pode usar os comandos da CLI para criar e gerenciar métricas personalizadas.

- criar-custom-metric
- describe-custom-metric

- list-custom-metrics
- atualização-métrica personalizada
- excluir-métrica personalizada
- list-security-profiles

APIs personalizadas

As APIs a seguir podem ser usadas para criar e gerenciar métricas personalizadas.

- CreateCustommetric
- DescribeCustomMetric
- ListCustomMetrics
- UpdateCustomMetric
- DeleteCustomMetric
- ListSecurityProfiles

Métricas do lado do dispositivo

Ao criar um perfil de segurança, você pode especificar o comportamento esperado do dispositivo IoT configurando comportamentos e limites para métricas geradas por dispositivos IoT. Veja a seguir as métricas do dispositivo, que são métricas de agentes instalados em seus dispositivos.

Saída de bytes (`aws:all-bytes-out`)

O número de bytes de saída de um dispositivo durante um determinado período.

Use esta métrica para especificar a quantidade máxima ou mínima de tráfego de saída que um dispositivo deve enviar, medido em bytes, em um determinado período.

Compatível com: Detecção de regras | ML Detect

Operadores: less-than | less-than-equals | greater-than | greater-than-equals

Valor: um inteiro não negativo.

Unidade: bytes

Duração: um inteiro não negativo. Os valores válidos são 300, 600, 900, 1800 ou 3.600 segundos.

Example

```
{
  "name": "TCP outbound traffic",
  "metric": "aws:all-bytes-out",
  "criteria": {
    "comparisonOperator": "less-than-equals",
    "value": {
      "count": 4096
    },
    "durationSeconds": 300,
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1
  },
  "suppressAlerts": true
}
```

Example Exemplo usando um **`statisticalThreshold`**.

```
{
  "name": "TCP outbound traffic",
  "metric": "aws:all-bytes-out",
  "criteria": {
    "comparisonOperator": "less-than-equals",
    "statisticalThreshold": {
      "statistic": "p50"
    },
    "durationSeconds": 900,
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1
  },
  "suppressAlerts": true
}
```

Example Exemplo usando ML Detect

```
{
  "name": "Outbound traffic ML behavior",
  "metric": "aws:all-bytes-out",
  "criteria": {
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1,
    "mlDetectionConfig": {
      "confidenceLevel": "HIGH"
    }
  },
  "suppressAlerts": true
}
```

Bytes em (`aws:all-bytes-in`)

O número de bytes de entrada para um dispositivo durante um determinado período.

Use esta métrica para especificar a quantidade máxima ou mínima de tráfego de entrada que um dispositivo deve receber, medido em bytes, em um determinado período.

Compatível com: Detecção de regras | ML Detect

Operadores: less-than | less-than-equals | greater-than | greater-than-equals

Valor: um inteiro não negativo.

Unidade: bytes

Duração: um inteiro não negativo. Os valores válidos são 300, 600, 900, 1800 ou 3.600 segundos.

Example

```
{
  "name": "TCP inbound traffic",
  "metric": "aws:all-bytes-in",
  "criteria": {
    "comparisonOperator": "less-than-equals",
    "value": {
      "count": 4096
    },
    "durationSeconds": 300,
    "consecutiveDatapointsToAlarm": 1,
```

```
    "consecutiveDatapointsToClear": 1
  },
  "suppressAlerts": true
}
```

Example Exemplo usando um **`statisticalThreshold`**.

```
{
  "name": "TCP inbound traffic",
  "metric": "aws:all-bytes-in",
  "criteria": {
    "comparisonOperator": "less-than-equals",
    "statisticalThreshold": {
      "statistic": "p90"
    },
    "durationSeconds": 300,
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1
  },
  "suppressAlerts": true
}
```

Example Exemplo usando ML Detect

```
{
  "name": "Inbound traffic ML behavior",
  "metric": "aws:all-bytes-in",
  "criteria": {
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1,
    "mlDetectionConfig": {
      "confidenceLevel": "HIGH"
    }
  },
  "suppressAlerts": true
}
```

Contagem de portas TCP escutando (`aws:num-listening-tcp-ports`)

O número de portas TCP nas quais o dispositivo está em escuta.

Use esta métrica para especificar o número máximo de portas TCP que cada dispositivo deve monitorar.

Compatível com: Detecção de regras | ML Detect

Unidade: falhas

Operadores: less-than | less-than-equals | greater-than | greater-than-equals

Valor: um inteiro não negativo.

Unidades: falhas

Duração: um inteiro não negativo. Os valores válidos são 300, 600, 900, 1800 ou 3.600 segundos.

Example

```
{
```

```
  "name": "Max TCP Ports",
  "metric": "aws:num-listening-tcp-ports",
  "criteria": {
    "comparisonOperator": "less-than-equals",
    "value": {
      "count": 5
    },
    "durationSeconds": 300,
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1
  },
  "suppressAlerts": true
}
```

Example Exemplo usando um **`statisticalThreshold`**.

```
{
  "name": "Max TCP Ports",
  "metric": "aws:num-listening-tcp-ports",
  "criteria": {
    "comparisonOperator": "less-than-equals",
    "statisticalThreshold": {
      "statistic": "p50"
    },
    "durationSeconds": 300,
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1
  },
  "suppressAlerts": true
}
```

Example Exemplo usando ML detect

```
{
  "name": "Max TCP Port ML behavior",
  "metric": "aws:num-listening-tcp-ports",
  "criteria": {
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1,
    "mlDetectionConfig": {
      "confidenceLevel": "HIGH"
    }
  },
  "suppressAlerts": true
}
```

Contagem de portas UDP escutando (`aws:num-listening-udp-ports`)

O número de portas UDP nas quais o dispositivo está em escuta.

Use esta métrica para especificar o número máximo de portas UDP que cada dispositivo deve monitorar.

Compatível com: Detecção de regras | ML Detect

Unidade: falhas

Operadores: less-than | less-than-equals | greater-than | greater-than-equals

Valor: um inteiro não negativo.

Unidades: falhas

Duração: um inteiro não negativo. Os valores válidos são 300, 600, 900, 1800 ou 3.600 segundos.

Example

```
{
  "name": "Max UDP Ports",
  "metric": "aws:num-listening-udp-ports",
  "criteria": {
    "comparisonOperator": "less-than-equals",
    "value": {
      "count": 5
    },
    "durationSeconds": 300,
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1
  },
  "suppressAlerts": true
}
```

Example Exemplo usando um **`statisticalThreshold`**.

```
{
  "name": "Max UDP Ports",
  "metric": "aws:num-listening-udp-ports",
  "criteria": {
    "comparisonOperator": "less-than-equals",
    "statisticalThreshold": {
      "statistic": "p50"
    },
    "durationSeconds": 300,
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1
  },
  "suppressAlerts": true
}
```

Example Exemplo usando ML Detect

```
{
  "name": "Max UPD Port ML behavior",
  "metric": "aws:num-listening-tcp-ports",
  "criteria": {
  "consecutiveDatapointsToAlarm": 1,
  "consecutiveDatapointsToClear": 1,
  "mlDetectionConfig": {
    "confidenceLevel": "HIGH"
   }
 },
  "suppressAlerts": true
}
```

Pacotes de saída (`aws:all-packets-out`)

O número de pacotes de saída de um dispositivo durante um determinado período.

Use esta métrica para especificar a quantidade máxima ou mínima do tráfego total de saída que um dispositivo deve enviar em um determinado período.

Compatível com: Detecção de regras | ML Detect

Operadores: less-than | less-than-equals | greater-than | greater-than-equals

Valor: um inteiro não negativo.

Unidades: pacotes

Duração: um inteiro não negativo. Os valores válidos são 300, 600, 900, 1800 ou 3.600 segundos.

Example

```
{
  "name": "TCP outbound traffic",
  "metric": "aws:all-packets-out",
  "criteria": {
    "comparisonOperator": "less-than-equals",
    "value": {
      "count": 100
    },
    "durationSeconds": 300,
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1
  },
  "suppressAlerts": true
}
```

Example Exemplo usando um **`statisticalThreshold`**.

```
{
  "name": "TCP outbound traffic",
  "metric": "aws:all-packets-out",
  "criteria": {
    "comparisonOperator": "less-than-equals",
    "statisticalThreshold": {
      "statistic": "p90"
    },
    "durationSeconds": 300,
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1
  },
  "suppressAlerts": true
}
```

Example Exemplo usando ML Detect

```
{
  "name": "Outbound sent ML behavior",
  "metric": "aws:all-packets-out",
  "criteria": {
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1,
    "mlDetectionConfig": {
      "confidenceLevel": "HIGH"
    }
  },
  "suppressAlerts": true
}
```

Pacotes em (`aws:all-packets-in`)

O número de pacotes de entrada para um dispositivo durante um determinado período.

Use esta métrica para especificar a quantidade máxima ou mínima do tráfego total de entrada que um dispositivo deve receber em um determinado período.

Compatível com: Detecção de regras | Detecção de ML

Operadores: less-than | less-than-equals | greater-than | greater-than-equals

Valor: um inteiro não negativo.

Unidades: pacotes

Duração: um inteiro não negativo. Os valores válidos são 300, 600, 900, 1800 ou 3.600 segundos.

Example

```
{
  "name": "TCP inbound traffic",
  "metric": "aws:all-packets-in",
  "criteria": {
    "comparisonOperator": "less-than-equals",
    "value": {
      "count": 100
    },
    "durationSeconds": 300,
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1
  },
  "suppressAlerts": true
}
```

Example

Exemplo usando um `statisticalThreshold`.

```
{
  "name": "TCP inbound traffic",
  "metric": "aws:all-packets-in",
  "criteria": {
    "comparisonOperator": "less-than-equals",
    "statisticalThreshold": {
      "statistic": "p90"
    },
    "durationSeconds": 300,
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1
  },
  "suppressAlerts": true
}
```

Example Exemplo usando ML Detect

```
{
  "name": "Inbound sent ML behavior",
  "metric": "aws:all-packets-in",
  "criteria": {
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1,
    "mlDetectionConfig": {
      "confidenceLevel": "HIGH"
    }
  },
```

```
  "suppressAlerts": true
}
```

IPs de destino (`aws:destination-ip-addresses`)

Um conjunto de destinos IP.

Use esta métrica para especificar um conjunto de roteamentos entre domínios (anteriormente chamado de lista de permissões) ou negado (anteriormente chamado de lista negra) em que cada dispositivo deve ou não se conectar aoAWS IoT.

Compatível com: Detectar regras

Operadores: in-cidr-set | not-in-cidr-set

Valores: uma lista de CIDRs

Unidades: n/a

Example

```
{
  "name": "Denied source IPs",
  "metric": "aws:source-ip-address",
  "criteria": {
    "comparisonOperator": "not-in-cidr-set",
    "value": {
      "cidrs": [ "12.8.0.0/16", "15.102.16.0/24" ]
    }
  },
  "suppressAlerts": true
}
```

Portas TCP de escuta (`aws:listening-tcp-ports`)

As portas TCP nas quais o dispositivo está em escuta.

Use esta métrica para especificar um conjunto de portas TCP permitidas (anteriormente chamado de lista de permissões) ou negadas (anteriormente chamado de lista negra) nas quais cada dispositivo deve ou não ouvir.

Compatível com: Detectar regras

Operadores: in-port-set | not-in-port-set

Valores: uma lista de portas

Unidades: n/a

Example

```
{
  "name": "Listening TCP Ports",
  "metric": "aws:listening-tcp-ports",
  "criteria": {
    "comparisonOperator": "in-port-set",
    "value": {
      "ports": [ 443, 80 ]
    }
```

```
  },
  "suppressAlerts": true
}
```

Portas UDP de escuta (`aws:listening-udp-ports`)

As portas UDP nas quais o dispositivo está em escuta.

Use esta métrica para especificar um conjunto de portas UDP permitidas (anteriormente chamado de lista de permissões) ou negadas (anteriormente chamado de lista negra) nas quais cada dispositivo deve ou não ouvir.

Compatível com: Detectar regras

Operadores: in-port-set | not-in-port-set

Valores: uma lista de portas

Unidades: n/a

Example

```
{
  "name": "Listening UDP Ports",
  "metric": "aws:listening-udp-ports",
  "criteria": {
    "comparisonOperator": "in-port-set",
    "value": {
      "ports": [ 1025, 2000 ]
    }
  }
}
```

Contagem de conexões TCP estabelecidas (`aws:num-established-tcp-connections`)

O número de conexões TCP para um dispositivo.

Use esta métrica para especificar o número máximo ou mínimo de conexões TCP ativas que cada dispositivo deve ter (Todos os estados TCP).

Compatível com: Detecção de regras | ML Detect

Operadores: less-than | less-than-equals | greater-than | greater-than-equals

Valor: um inteiro não negativo.

Unidades: conexões

Example

```
{
  "name": "TCP Connection Count",
  "metric": "aws:num-established-tcp-connections",
  "criteria": {
    "comparisonOperator": "less-than-equals",
    "value": {
      "count": 3
    },
```

```
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1
  },
  "suppressAlerts": true
}
```

Example Exemplo usando um **`statisticalThreshold`**.

```
{
  "name": "TCP Connection Count",
  "metric": "aws:num-established-tcp-connections",
  "criteria": {
    "comparisonOperator": "less-than-equals",
    "statisticalThreshold": {
      "statistic": "p90"
    },
    "durationSeconds": 900,
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1
  },
  "suppressAlerts": true
}
```

Example Exemplo usando ML Detect

```
{
  "name": "Connection count ML behavior",
  "metric": "aws:num-established-tcp-connections",
  "criteria": {
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1,
    "mlDetectionConfig": {
      "confidenceLevel": "HIGH"
    }
  },
  "suppressAlerts": true
}
```

Especificação da documentação de métricas do dispositivo

Estrutura geral

Nome longo	Nome curto	Obrigatório	Type	Restrições	Observações
header	hed	Y	Objeto		Bloco completo necessário para um relatório bem formado.
métricas	met	Y	Objeto		Um relatório pode ter ambos ou pelo menos um`metricsou``custom_metr`
custom_metrics	cmet	Y	Objeto		Um relatório pode ter ambos ou pelo menos um`metricsou``custom_metr`

Bloco do cabeçalho

Nome longo	Nome curto	Obrigatório	Type	Restrições	Observações
report_id	rid	Y	Inteiro		Aumentando o valor de forma monotônica. Timestamp epoch recomendado.
versão	v	Y	String	Major.Minor	Pequenos incrementos com adição de campo. Principais incrementos se as métricas forem removidas.

Bloco de métricas:

Conexões TCP

Nome longo	Nome curto	Elemento principal	Obrigatório	Type	Restrições	Observações
tcp_connections	tc	métricas	N	Objeto		
established_connections	ec	tcp_connections	N	Objeto		Estado TCP estabelecido
connections	cs	established_connections	N	List<Object>		
remote_addr	rad	connections	Y	telefone	ip:port	IP pode ser IPv6 ou IPv4
local_port	lp	connections	N	telefone	>= 0	
local_interface	li	connections	N	String		Nome da interface
total	t	established_connections	N	telefone	>= 0	Número de conexões estabelecidas

Portas TCP de escuta

Nome longo	Nome curto	Elemento principal	Obrigatório	Type	Restrições	Observações
listening_tcp_ports	lpts	métricas	N	Objeto		
ports	pts	listening_tcp_ports	N	List<Object>	> 0	
port	pt	ports	N	telefone	> 0	as portas devem ser

Nome longo	Nome curto	Elemento principal	Obrigatório	Type	Restrições	Observações
						números maiores que 0
interface	se	ports	N	String		Nome da interface
total	t	listening_tcp_ports	N	telefone	>= 0	

Portas UDP de escuta

Nome longo	Nome curto	Elemento principal	Obrigatório	Type	Restrições	Observações
listening_udp_ports	pts	métricas	N	Objeto		
ports	pts	listening_udp_ports	N	List<Port>	> 0	
port	pt	ports	N	telefone	> 0	As portas devem ser números maiores que 0
interface	se	ports	N	String		Nome da interface
total	t	listening_udp_ports	N	telefone	>= 0	

Estatísticas de rede

Nome longo	Nome curto	Elemento principal	Obrigatório	Type	Restrições	Observações
network_stats	ns	métricas	N	Objeto		
bytes_in	bi	network_stats	N	telefone	Métrica delta, >= 0	
bytes_out	bo	network_stats	N	telefone	Métrica delta, >= 0	
packets_in	pi	network_stats	N	telefone	Métrica delta, >= 0	
packets_out	po	network_stats	N	telefone	Métrica delta, >= 0	

Example

A seguinte estrutura JSON usa nomes longos.

```
{
  "header": {
```

```
    "report_id": 1530304554,
    "version": "1.0"
  },
  "metrics": {
    "listening_tcp_ports": {
      "ports": [
        {
          "interface": "eth0",
          "port": 24800
        },
        {
          "interface": "eth0",
          "port": 22
        },
        {
          "interface": "eth0",
          "port": 53
        }
      ],
      "total": 3
    },
    "listening_udp_ports": {
      "ports": [
        {
          "interface": "eth0",
          "port": 5353
        },
        {
          "interface": "eth0",
          "port": 67
        }
      ],
      "total": 2
    },
    "network_stats": {
      "bytes_in": 29358693495,
      "bytes_out": 26485035,
      "packets_in": 10013573555,
      "packets_out": 11382615
    },
    "tcp_connections": {
      "established_connections": {
        "connections": [
          {
            "local_interface": "eth0",
            "local_port": 80,
            "remote_addr": "192.168.0.1:8000"
          },
          {
            "local_interface": "eth0",
            "local_port": 80,
            "remote_addr": "192.168.0.1:8000"
          }
        ],
        "total": 2
      }
    }
  },
  "custom_metrics": {
    "MyMetricOfType_Number": [
      {
        "number": 1
      }
    ],
    "MyMetricOfType_NumberList": [
      {
```

```
        "number_list": [
          1,
          2,
          3
        ]
      }
    ],
    "MyMetricOfType_StringList": [
      {
        "string_list": [
          "value_1",
          "value_2"
        ]
      }
    ],
    "MyMetricOfType_IpList": [
      {
        "ip_list": [
          "172.0.0.0",
          "172.0.0.10"
        ]
      }
    ]
  }
}
```

Example Exemplo de estrutura JSON usando nomes curtos

```
{
  "hed": {
    "rid": 1530305228,
    "v": "1.0"
  },
  "met": {
    "tp": {
      "pts": [
        {
          "if": "eth0",
          "pt": 24800
        },
        {
          "if": "eth0",
          "pt": 22
        },
        {
          "if": "eth0",
          "pt": 53
        }
      ],
      "t": 3
    },
    "up": {
      "pts": [
        {
          "if": "eth0",
          "pt": 5353
        },
        {
          "if": "eth0",
          "pt": 67
        }
      ],
      "t": 2
    },
```

```
    "ns": {
      "bi": 29359307173,
      "bo": 26490711,
      "pi": 10014614051,
      "po": 11387620
    },
    "tc": {
      "ec": {
        "cs": [
          {
            "li": "eth0",
            "lp": 80,
            "rad": "192.168.0.1:8000"
          },
          {
            "li": "eth0",
            "lp": 80,
            "rad": "192.168.0.1:8000"
          }
        ],
        "t": 2
      }
    }
  },
  "cmet": {
    "MyMetricOfType_Number": [
      {
        "number": 1
      }
    ],
    "MyMetricOfType_NumberList": [
      {
        "number_list": [
          1,
          2,
          3
        ]
      }
    ],
    "MyMetricOfType_StringList": [
      {
        "string_list": [
          "value_1",
          "value_2"
        ]
      }
    ],
    "MyMetricOfType_IpList": [
      {
        "ip_list": [
          "172.0.0.0",
          "172.0.0.10"
        ]
      }
    ]
  }
}
```

Envio de métricas de dispositivos

O AWS IoT Device Defender Detect pode coletar, agregar e monitorar dados de métricas gerados por dispositivos da AWS IoT para identificar dispositivos que apresentam comportamento anormal. Esta seção mostra como enviar métricas de um dispositivo para o AWS IoT Device Defender.

Você deve implantar com segurança oAWS IoTSDK versão dois em seuAWS IoTDispositivos conectados ou gateways de dispositivos para coletar métricas do dispositivo. Veja a lista completa de SDKsaqui.

Você pode usarAWS IoTDevice Client para publicar métricas, pois fornece um único agente que cobre os recursos presentes em ambosAWS IoT Device DefendereAWS IoTGerenciamento de dispositivos. Esses recursos incluem trabalhos, tunelamento seguro,AWS IoT Device Defenderpublicação de métricas e muito mais.

Você publica métricas do lado do dispositivo noTópico reservadoemAWS IoTpeloAWS IoT Device Defenderpara coletar e avaliar.

Usar oAWS IoTDevice Client para publicar métricas

Para instalar oAWS IoTDevice Client, você pode fazer download doGithub. Depois de instalar oAWS IoTDevice Client no dispositivo para o qual você deseja coletar dados do lado do dispositivo, você deve configurá-lo para enviar métricas do lado do dispositivo paraAWS IoT Device Defender. Verifique se oAWS IoTCliente de dispositivoArquivo de configuráotem os parâmetros a seguir definidos na`device-defender`seção :

```
"device-defender":    {
      "enabled":    true,
      "interval-in-seconds": 300
  }
```

> **Warning**
>
> Você deve definir o intervalo de tempo para um mínimo de 300 segundos. Se você definir o intervalo de tempo para menos de 300 segundos, seus dados métricos podem ser limitados.

Depois de atualizar sua configuração, você pode criar perfis e comportamentos de segurança naAWS IoT Device Defenderconsole para monitorar as métricas que seus dispositivos publicam na nuvem. Você pode encontrar métricas publicadas noAWS IoT Coreconsole escolhendo Defender, Detectar e, em seguida, Métricas.

Métricas do lado da nuvem

Ao criar um perfil de segurança, você pode especificar o comportamento esperado do dispositivo IoT configurando comportamentos e limites para métricas geradas por dispositivos IoT. Veja a seguir métricas da nuvem, que são métricas daAWS IoT.

Tamanho da mensagem (aws:message-byte-size)

O número de bytes na fila. Use esta métrica para especificar o tamanho máximo ou mínimo (em bytes) de cada mensagem transmitida a partir de um dispositivo para a AWS IoT.

Compatível com: Detecção de regras | ML Detect

Operadores: less-than | less-than-equals | greater-than | greater-than-equals

Valor: um inteiro não negativo.

Unidade: bytes

Example

```
{
  "name": "Max Message Size",
  "metric": "aws:message-byte-size",
  "criteria": {
    "comparisonOperator": "less-than-equals",
```

```
    "value": {
      "count": 1024
    },
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1
  },
  "suppressAlerts": true
}
```

Example Exemplo usando um **`statisticalThreshold`**.

```
{

  "name": "Large Message Size",
  "metric": "aws:message-byte-size",
  "criteria": {
    "comparisonOperator": "less-than-equals",
    "statisticalThreshold": {
      "statistic": "p90"
    },
    "durationSeconds": 300,
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1
  },
  "suppressAlerts": true
}
```

Example Exemplo usando ML Detect

```
{
  "name": "Message size ML behavior",
  "metric": "aws:message-byte-size",
  "criteria": {
  "consecutiveDatapointsToAlarm": 1,
  "consecutiveDatapointsToClear": 1,
  "mlDetectionConfig": {
    "confidenceLevel": "HIGH"
   }
 },
  "suppressAlerts": true
}
```

Um alarme será emitido para um dispositivo se durante três períodos consecutivos de cinco minutos, ele transmitir mensagens em que o tamanho cumulativo é superior ao medido para 90% de todos os outros dispositivos que relatam esse comportamento do Perfil de Segurança.

Mensagens enviadas (aws:num-messages-sent)

O número de mensagens enviadas por um dispositivo durante um determinado período.

Use esta métrica para especificar o número máximo ou mínimo de mensagens que podem ser enviadas entreAWS IoTe cada dispositivo em um determinado período.

Compatível com: Detecção de regras | ML Detect

Operadores: less-than | less-than-equals | greater-than | greater-than-equals

Valor: um inteiro não negativo.

Unidades: mensagens

Duração: um inteiro não negativo. Os valores válidos são 300, 600, 900, 1800 ou 3.600 segundos.

Example

```
{

  "name": "Out bound message count",
  "metric": "aws:num-messages-sent",
  "criteria": {
    "comparisonOperator": "less-than-equals",
    "value": {
      "count": 50
    },
    "durationSeconds": 300,
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1
    },
  "suppressAlerts": true
}
```

Example Exemplo usando um **`statisticalThreshold`**.

```
{

  "name": "Out bound message rate",
  "metric": "aws:num-messages-sent",
  "criteria": {
    "comparisonOperator": "less-than-equals",
    "statisticalThreshold": {
      "statistic": "p99"
    },
    "durationSeconds": 300,
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1
  },
  "suppressAlerts": true
}
```

Example Exemplo usando ML Detect

```
{
  "name": "Messages sent ML behavior",
  "metric": "aws:num-messages-sent",
  "criteria": {
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1,
    "mlDetectionConfig": {
      "confidenceLevel": "HIGH"
    }
  },
  "suppressAlerts": true
}
```

Mensagens recebidas (aws:num-messages-received)

O número de mensagens recebidas por um dispositivo durante um determinado período.

Use esta métrica para especificar o número máximo ou mínimo de mensagens que podem ser recebidas entreAWS IoTe cada dispositivo em um determinado período.

Compatível com: Detecção de regras | ML Detect

Operadores: less-than | less-than-equals | greater-than | greater-than-equals

Valor: um inteiro não negativo.

Unidades: mensagens

Duração: um inteiro não negativo. Os valores válidos são 300, 600, 900, 1800 ou 3.600 segundos.

Example

```
{
  "name": "In bound message count",
  "metric": "aws:num-messages-received",
  "criteria": {
    "comparisonOperator": "less-than-equals",
    "value": {
      "count": 50
    },
    "durationSeconds": 300,
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1
    },
  "suppressAlerts": true
}
```

Example Exemplo usando um **`statisticalThreshold`**.

```
{
  "name": "In bound message rate",
  "metric": "aws:num-messages-received",
  "criteria": {
    "comparisonOperator": "less-than-equals",
    "statisticalThreshold": {
      "statistic": "p99"
    },
    "durationSeconds": 300,
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1
  },
  "suppressAlerts": true
}
```

Example Exemplo usando ML Detect

```
{
  "name": "Messages received ML behavior",
  "metric": "aws:num-messages-received",
  "criteria": {
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1,
    "mlDetectionConfig": {
      "confidenceLevel": "HIGH"
    }
  },
  "suppressAlerts": true
}
```

Falhas de autorização (aws:num-authorization-failures)

Use esta métrica para especificar o número máximo de falhas de autorização permitidas para cada dispositivo em um determinado período. Uma falha de autorização ocorre quando uma solicitação de um dispositivo para a AWS IoT é negada (por exemplo, se um dispositivo tenta publicar em um tópico para o qual não tem permissões suficientes).

Compatível com: Detecção de regras | ML Detect

Unidade: falhas

Operadores: less-than | less-than-equals | greater-than | greater-than-equals

Valor: um inteiro não negativo.

Duração: um inteiro não negativo. Os valores válidos são 300, 600, 900, 1800 ou 3.600 segundos.

Example

```
{
  "name": "Authorization Failures",
  "metric": "aws:num-authorization-failures",
  "criteria": {
    "comparisonOperator": "less-than",
    "value": {
      "count": 5
    },
    "durationSeconds": 300,
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1
  },
  "suppressAlerts": true
}
```

Example Exemplo usando um **`statisticalThreshold`**.

```
{
  "name": "Authorization Failures",
  "metric": "aws:num-authorization-failures",
  "criteria": {
    "comparisonOperator": "less-than-equals",
    "statisticalThreshold": {
      "statistic": "p50"
    },
    "durationSeconds": 300,
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1
  },
  "suppressAlerts": true
}
```

Example Exemplo usando ML Detect

```
{
  "name": "Authorization failures ML behavior",
  "metric": "aws:num-authorization-failures",
  "criteria": {
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1,
    "mlDetectionConfig": {
      "confidenceLevel": "HIGH"
    }
  },
  "suppressAlerts": true
}
```

IP de origem (aws:source-ip-address)

O endereço IP a partir do qual um dispositivo se conectou à AWS IoT.

Use esta métrica para especificar um conjunto de roteamentos entre domínios (anteriormente chamado de lista de permissões) ou negados (anteriormente chamado de lista negra) dos quais cada dispositivo deve ou não se conectar aoAWS IoT.

Compatível com: Detecção de regras

Operadores: in-cidr-set | not-in-cidr-set

Valores: uma lista de CIDRs

Unidades: n/a

Example

```
{
  "name": "Denied source IPs",
  "metric": "aws:source-ip-address",
  "criteria": {
    "comparisonOperator": "not-in-cidr-set",
    "value": {
      "cidrs": [ "12.8.0.0/16", "15.102.16.0/24" ]
    }
  },
  "suppressAlerts": true
}
```

Tentativas de conexão (aws:num-connection-tentativas)

O número de vezes que um dispositivo tentou fazer uma conexão em um determinado período.

Use esta métrica para especificar o número máximo ou mínimo de tentativas de conexão para cada dispositivo. Tanto as tentativas bem-sucedidas como as malsucedidas são contadas.

Compatível com: Detecção de regras | ML Detect

Operadores: less-than | less-than-equals | greater-than | greater-than-equals

Valor: um inteiro não negativo.

Unidades: tentativas de conexão

Duração: um inteiro não negativo. Os valores válidos são 300, 600, 900, 1800 ou 3.600 segundos.

Example

```
{
  "name": "Connection Attempts",
  "metric": "aws:num-connection-attempts",
  "criteria": {
    "comparisonOperator": "less-than-equals",
    "value": {
      "count": 5
    },
    "durationSeconds": 600,
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1
  },
  "suppressAlerts": true
}
```

Example Exemplo usando um **`statisticalThreshold`**.

```
{
  "name": "Connection Attempts",
  "metric": "aws:num-connection-attempts",
  "criteria": {
    "comparisonOperator": "less-than-equals",
    "statisticalThreshold": {
      "statistic": "p10"
    },
    "durationSeconds": 300,
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1
  },
  "suppressAlerts": true
}
```

Example Exemplo usando ML Detect

```
{
  "name": "Connection attempts ML behavior",
  "metric": "aws:num-connection-attempts",
  "criteria": {
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1,
    "mlDetectionConfig": {
      "confidenceLevel": "HIGH"
    }
  },
  "suppressAlerts": false
}
```

Desconecta (aws:num-disconnect)

O número de vezes que um dispositivo foi desconectado da AWS IoT durante determinado período.

Use esta métrica para especificar o número máximo ou mínimo de vezes que um dispositivo foi desconectado da AWS IoT durante determinado período.

Compatível com: Detecção de regras | ML Detect

Operadores: less-than | less-than-equals | greater-than | greater-than-equals

Valor: um inteiro não negativo.

Unidades: desconexões

Duração: um inteiro não negativo. Os valores válidos são 300, 600, 900, 1800 ou 3.600 segundos.

Example

```
{
  "name": "Disconnections",
  "metric": "aws:num-disconnects",
  "criteria": {
    "comparisonOperator": "less-than-equals",
    "value": {
      "count": 5
    },
    "durationSeconds": 600,
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1
```

```
  },
  "suppressAlerts": true
}
```

Example Exemplo usando um **`statisticalThreshold`**.

```
{
  "name": "Disconnections",
  "metric": "aws:num-disconnects",
  "criteria": {
    "comparisonOperator": "less-than-equals",
    "statisticalThreshold": {
      "statistic": "p10"
    },
    "durationSeconds": 300,
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1
  },
  "suppressAlerts": true
}
```

Example Exemplo usando ML Detect

```
{
  "name": "Disconnects ML behavior",
  "metric": "aws:num-disconnects",
  "criteria": {
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1,
    "mlDetectionConfig": {
      "confidenceLevel": "HIGH"
    }
  },
  "suppressAlerts": true
}
```

Escopo de métricas em perfis de segurança usando dimensões

Dimensões são atributos que podem ser definidos para obter dados mais precisos sobre métricas e comportamentos no perfil de segurança. Defina o escopo fornecendo um valor ou padrão que é usado como um filtro. Por exemplo, é possível definir uma dimensão de filtro de tópico que aplica uma métrica somente a tópicos MQTT que correspondam a um valor determinado, como "data/bulb/+/activity". Para obter informações sobre como definir uma dimensão que você pode usar em seu perfil de segurança, consulteCreateDimension.

Os valores de dimensão oferecem suporte a curingas do MQTT. Os curingas do MQTT ajudam você a se inscrever em vários tópicos simultaneamente. Existem dois tipos diferentes de curingas: nível único (`+`) e vários níveis (`#`). Por exemplo, o valor da dimensão `Data/bulb/+/activity` cria uma inscrição que corresponde a todos os tópicos existentes no mesmo nível que `+`. Os valores de dimensão também oferecem suporte à variável de substituição de ID de cliente MQTT $ {iot:ClientId}.

As dimensões do tipo TOPIC_FILTER são compatíveis com o seguinte conjunto de métricas da nuvem:

- Número de mensagens enviadas
- Número de mensagens recebidas
- Tamanho do byte da mensagem
- Endereço IP de origem

- Número de falhas de autorização

Como usar dimensões no console

Como criar e aplicar uma dimensão a um comportamento de perfil de segurança

1. NoAWS IoTconsoleExpandir no painel de Navegação, expandaDefenderExpandirDetectarEscolha e, depois, escolhaPerfis de segurança.
2. Na página Security profiles (Perfis de segurança), escolha Create (Criar) para adicionar um novo perfil de segurança ou Edit (Editar) para aplicar uma dimensão a um perfil de segurança existente.
3. Na página Expected behaviors (Comportamentos esperados), selecione uma das cinco dimensões métricas da nuvem compatíveis em Metric (Métrica). As caixas Dimension (Dimensão) e Dimension operator (Operador de dimensão) são exibidas. Para obter informações sobre métricas da nuvem compatíveis, consulte Escopo de métricas em perfis de segurança usando dimensões (p. 1035).
4. para oDimensão, escolhaAdicionar dimensão.
5. Na página Create a new dimension (Criar uma nova dimensão), insira detalhes para sua nova dimensão. Dimensions values (Valores de dimensões) permite caracteres curinga MQTT # e + e a variável de substituição de ID de cliente MQTT ${iot:ClientId}.

6. Escolha Save (Salvar).
7. Opcionalmente, adicione dimensões a métricas emMétricas adicionais para manter.
8. Para concluir a criação do comportamento, digite as informações nos outros campos obrigatórios e, depois, escolhaPróximo.
9. Conclua as etapas restantes para concluir a criação de um perfil de segurança.

Como visualizar suas violações

1. NoAWS IoTconsoleExpandir no painel de Navegação, expandaDefenderExpandirDetectarEscolha e, depois, escolhaViolações.

2. NoComportamentoColuna, pause sobre o comportamento para o qual você deseja ver as informações de violação.

Como exibir e atualizar suas dimensões

1. NoAWS IoTconsoleExpandir no painel de Navegação, expandaDefenderExpandirDetectarEscolha e, depois, escolhaDimensões.
2. Selecione a dimensão que deseja editar.
3. Escolha Ações e, em seguida, escolha Editar.

Como excluir uma dimensão

1. NoAWS IoTconsoleExpandir no painel de Navegação, expandaDefenderExpandirDetectarEscolha e, depois, escolhaDimensões.
2. Selecione a dimensão que deseja excluir.
3. Confirme se a dimensão não está anexada a um perfil de segurança marcando a coluna Used in (Usado em). Se a dimensão estiver anexada a um perfil de segurança, abra a página Security

profiles (Perfis de segurança) à esquerda e edite os perfis de segurança aos quais a dimensão está anexada. Quando exclui a dimensão, você também exclui o comportamento. Se quiser manter o comportamento, escolha as reticências e escolha Copy (Copiar). Depois, você pode prosseguir com a exclusão do comportamento. Se desejar excluir outra dimensão, siga as etapas nesta seção.

4. Escolha Actions (Ações) e, em seguida, escolha Delete (Excluir).

Como usar dimensões noAWS CLI

Como criar e aplicar uma dimensão a um comportamento de perfil de segurança

1. Primeiro crie a dimensão antes de anexá-la a um perfil de segurança. Usar aCreateDimensionComando para criar uma dimensão:

```
aws iot create-dimension \
  --name TopicFilterForAuthMessages \
  --type TOPIC_FILTER \
  --string-values device/+/auth
```

A saída desse comando é semelhante à seguinte:

```
{
```

```
    "arn": "arn:aws:iot:us-west-2:123456789012:dimension/TopicFilterForAuthMessages",
    "name": "TopicFilterForAuthMessages"
}
```

2. Adicione a dimensão a um perfil de segurança existente usandoUpdateSecurityProfileAdicione a dimensão a um novo perfil de segurança usandoCreateSecurityProfile. No exemplo a seguir, criamos um perfil de segurança que verifica se as mensagens para `TopicFilterForAuthMessages` estão abaixo de 128 bytes e que retém o número de mensagens enviadas para tópicos não auth.

```
aws iot create-security-profile \
  --security-profile-name ProfileForConnectedDevice \
  --security-profile-description "Check to see if messages to
 TopicFilterForAuthMessages are under 128 bytes and retains the number of messages sent
 to non-auth topics." \
  --behaviors "[{\"name\":\"CellularBandwidth\",\"metric\":\"aws:message-byte-size
\",\"criteria\":{\"comparisonOperator\":\"less-than\",\"value\":{\"count\":128},
\"consecutiveDatapointsToAlarm\":1,\"consecutiveDatapointsToClear\":1}},{\"name
\":\"Authorization\",\"metric\":\"aws:num-authorization-failures\",\"criteria\":
{\"comparisonOperator\":\"less-than\",\"value\":{\"count\":10},\"durationSeconds\":300,
\"consecutiveDatapointsToAlarm\":1,\"consecutiveDatapointsToClear\":1}}]" \
  --additional-metrics-to-retain-v2 "[{\"metric\": \"aws:num-authorization-failures\",
\"metricDimension\": {\"dimensionName\": \"TopicFilterForAuthMessages\",\"operator\":
 \"NOT_IN\"}}]"
```

A saída desse comando é semelhante à seguinte:

```
{
    "securityProfileArn": "arn:aws:iot:us-west-2:1234564789012:securityprofile/
ProfileForConnectedDevice",
    "securityProfileName": "ProfileForConnectedDevice"
}
```

Para economizar tempo, você também pode carregar um parâmetro a partir de um arquivo em vez de digitá-lo como um valor de parâmetro de linha de comando. Para obter mais informações, consultecarregarAWS CLIParâmetros de um arquivo. Veja a seguir o parâmetro `behavior` no formato JSON expandido:

```
[
  {
    "criteria": {
      "comparisonOperator": "less-than",
      "consecutiveDatapointsToAlarm": 1,
      "consecutiveDatapointsToClear": 1,
      "value": {
        "count": 128
      }
    },
    "metric": "aws:message-byte-size",
    "metricDimension": {
      "dimensionName:": "TopicFilterForAuthMessages"
    },
    "name": "CellularBandwidth"
  }
]
```

Como visualizar perfis de segurança com uma dimensão

- Usar aListSecurityProfilesComando para visualizar perfis de segurança com determinada dimensão:

```
aws iot list-security-profiles \
  --dimension-name TopicFilterForAuthMessages
```

A saída desse comando é semelhante à seguinte:

```
{
    "securityProfileIdentifiers": [
        {
            "name": "ProfileForConnectedDevice",
            "arn": "arn:aws:iot:us-west-2:1234564789012:securityprofile/
ProfileForConnectedDevice"
        }
    ]
}
```

Como atualizar a dimensão

- Usar aUpdateDimensionComando para atualizar uma dimensão:

```
aws iot update-dimension \
  --name TopicFilterForAuthMessages \
  --string-values device/${iot:ClientId}/auth
```

A saída desse comando é semelhante à seguinte:

```
{
    "name": "TopicFilterForAuthMessages",
    "lastModifiedDate": 1585866222.317,
    "stringValues": [
        "device/${iot:ClientId}/auth"
    ],
    "creationDate": 1585854500.474,
    "type": "TOPIC_FILTER",
    "arn": "arn:aws:iot:us-west-2:1234564789012:dimension/TopicFilterForAuthMessages"
}
```

Como excluir uma dimensão

1. Para excluir uma dimensão, primeiro desanexe-a de todos os perfis de segurança aos quais ela está anexada. Usar aListSecurityProfilesComando para visualizar perfis de segurança com determinada dimensão.
2. Para remover uma dimensão de um perfil de segurança, use aUpdateSecurityProfilecomando. Insira todas as informações que deseja manter, mas exclua a dimensão:

```
aws iot update-security-profile \
  --security-profile-name ProfileForConnectedDevice \
  --security-profile-description "Check to see if authorization fails 10 times in 5
 minutes or if cellular bandwidth exceeds 128" \
  --behaviors "[{\"name\":\"metric\":\"aws:message-byte-size\",\"criteria
\":{\"comparisonOperator\":\"less-than\",\"value\":{\"count\":128},
\"consecutiveDatapointsToAlarm\":1,\"consecutiveDatapointsToClear\":1}},{\"name
\":\"Authorization\",\"metric\":\"aws:num-authorization-failures\",\"criteria\":
{\comparisonOperator\":\"less-than\",\"value\"{\"count\":10},\"durationSeconds\":300,
\"consecutiveDatapointsToAlarm\":1,\"consecutiveDatapointsToClear\":1}}]"
```

A saída desse comando é semelhante à seguinte:

```
{
  "behaviors": [
    {
      "metric": "aws:message-byte-size",
      "name": "CellularBandwidth",
      "criteria": {
        "consecutiveDatapointsToClear": 1,
        "comparisonOperator": "less-than",
        "consecutiveDatapointsToAlarm": 1,
        "value": {
          "count": 128
        }
      }
    },
    {
      "metric": "aws:num-authorization-failures",
      "name": "Authorization",
      "criteria": {
        "durationSeconds": 300,
        "comparisonOperator": "less-than",
        "consecutiveDatapointsToClear": 1,
        "consecutiveDatapointsToAlarm": 1,
        "value": {
          "count": 10
        }
      }
    }
  ],
  "securityProfileName": "ProfileForConnectedDevice",
  "lastModifiedDate": 1585936349.12,
  "securityProfileDescription": "Check to see if authorization fails 10 times in 5
 minutes or if cellular bandwidth exceeds 128",
  "version": 2,
  "securityProfileArn": "arn:aws:iot:us-west-2:123456789012:securityprofile/Preo/
ProfileForConnectedDevice",
  "creationDate": 1585846909.127
}
```

3. Depois que a dimensão for desanexada, use aDeleteDimensionComando para excluir a dimensão:

```
aws iot delete-dimension \
  --name TopicFilterForAuthMessages
```

Permissões

Esta seção contém informações sobre como configurar as políticas e funções do IAM necessárias para gerenciarAWS IoT Device DefenderDetectar. Para obter mais informações, consulte o manual do usuário do IAM.

FornecerAWS IoT Device DefenderDetectar permissão para publicar alarmes em um tópico do SNS

Se você usar `alertTargets`Parâmetro emCreateSecurityProfile, você deverá especificar uma função do IAM com duas políticas: uma política de permissões e uma política de confiança. A política de permissões concede permissão para o AWS IoT Device Defender publicar notificações no tópico do SNS. A política de confiança concede ao AWS IoT Device Defender permissão para assumir a função necessária.

Política de permissão

```
{
  "Version":"2012-10-17",
  "Statement":[
      {
        "Effect":"Allow",
        "Action":[
          "sns:Publish"
        ],
        "Resource":[
          "arn:aws:sns:region:account-id:your-topic-name"
        ]
    }
  ]
}
```

Política de confiança

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

Transmitir política da função

Você também precisa de uma política de permissões anexada ao usuário do IAM que permita ao usuário transmitir funções. Consulte Conceder permissões ao usuário para transmitir uma função para um serviço da AWS.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Action": [
          "iam:GetRole",
          "iam:PassRole"
      ],
      "Resource": "arn:aws:iam::account-id:role/Role_To_Pass"
    }
  ]
}
```

Comandos de detecção

Você pode usar os comandos de detecção nesta seção para configurar o ML de detecção ou de regras de detecção de perfis de segurança para identificar e monitorar comportamentos incomuns que podem indicar um dispositivo comprometido.

Comandos de ação de detecção

Iniciar e gerenciar a execução do
CancelDetectMitigationActionsTask
DescribeDetectMitigationActionsTask
ListDetectMitigationActionsTasks
StartDetectMitigationActionsTask
ListDetectMitigationActionSextecutions

Comandos de ação de dimensão

Iniciar e gerenciar a execução do Dimensão
CreateDimension
DescribeDimension
ListDimensions
DeleteDimension
UpdateDimension

Comandos de ação customMetric

Iniciar e gerenciar CustomMetric execução
CreateCustommetric
UpdateCustomMetric
DescribeCustomMetric
ListCustomMetrics
DeleteCustomMetric

Comandos de ação do perfil de

Iniciar e gerenciar a execução do perfil de segurança
CreateSecurityProfile
AttachSecurityProfile
DetachSecurityProfile
DeleteSecurityProfile
DescribeSecurityProfile
ListTargetsForSecurityProfile
UpdateSecurityProfile

Iniciar e gerenciar a execução do perfil de segurança
ValidateSecurityProfileBehaviors
ListSecurityProfilesForTarget

Comandos de ação do alarme

Gerenciar alarmes e alvos
ListActiveViolations
ListViolationEvents
PutVerificationStateOnViolation

Comandos de ação ML Detect

Listar dados de treinamento do modelo de ML
GetBehaviorModelTrainingSummaries

Como usar o AWS IoT Device Defender Detect

1. Você pode usar o AWS IoT Device Defender Detect apenas com métricas no lado da nuvem, mas se planeja usar métricas informadas pelo dispositivo, primeiro é necessário implantar o SDK da AWS IoT nos dispositivos conectados da AWS IoT ou gateways de dispositivo. Para obter mais informações, consulteEnvio de métricas de dispositivos (p. 1027)
2. Considere visualizar as métricas que os dispositivos geram antes de definir comportamentos e criar alertas. A AWS IoT pode coletar métricas dos dispositivos para que você possa primeiro identificar comportamentos comuns ou incomuns em um grupo de dispositivos ou em todos os dispositivos da sua conta. Usar oCreateSecurityProfile, mas especifique apenas aquelesadditionalMetricsToRetainQue esteja interessado. Não especifiquebehaviorsNeste ponto.

   Use o console da AWS IoT para ver as métricas de dispositivo para ver o que constitui um comportamento típico para seus dispositivos.
3. Crie um conjunto de comportamentos para seu perfil de segurança. Comportamentos contêm métricas que especificam um comportamento normal para um grupo de dispositivos ou para todos os dispositivos em sua conta. Para obter mais informações e exemplos, consulte Métricas do lado da nuvem (p. 1028) e Métricas do lado do dispositivo (p. 1013). Depois de criar um conjunto de comportamentos, é possível validá-los comValidateSecurityProfileBehaviors.
4. Usar aCreateSecurityProfilePara criar um perfil de segurança que inclua os comportamentos. Você pode usar `oalertTargets`Para fazer com que alarmes sejam enviados para um destino (um tópico SNS) quando um dispositivo violar um comportamento. (Se você enviar alarmes usando o SNS lembre-se de que eles são contabilizados noConta da AWScota de tópico SNS. Uma grande explosão de violações pode exceder sua cota de tópicos do SNS. Você também pode usar CloudWatch Métricas para verificar a existência de violações. Para obter mais informações, consulteUsar métricas do AWS IoT (p. 439)
5. Usar aAttachSecurityProfilePara anexar o perfil de segurança a um grupo de dispositivos (um grupo de coisas), todas as coisas registradas em sua conta, todas as coisas não registradas ou todos os dispositivos.AWS IoT Device Defender O Detect inicia a verificação de comportamento anormal e, se forem detectadas violações de comportamento, envia alarmes. Você pode querer anexar um perfil de

segurança a todas as coisas não registradas se, por exemplo, você espera interagir com dispositivos móveis que não estão no registro de coisas de sua conta. Você pode definir diferentes conjuntos de comportamentos para diferentes grupos de dispositivos para atender às suas necessidades.

Para anexar um perfil de segurança a um grupo de dispositivos, é necessário especificar o ARN do grupo de coisas que os contém. O ARN de um grupo de coisas tem o seguinte formato.

```
arn:aws:iot:region:account-id:thinggroup/thing-group-name
```

Para anexar um perfil de segurança a todas as coisas registradas em umConta da AWS(ignorando coisas não registradas), você deve especificar um ARN com o seguinte formato.

```
arn:aws:iot:region:account-id:all/registered-things
```

Para anexar um perfil de segurança a todas as coisas não registradas, você deve especificar um ARN com o seguinte formato.

```
arn:aws:iot:region:account-id:all/unregistered-things
```

Para anexar um perfil de segurança a todos os dispositivos, você deve especificar um ARN com o seguinte formato.

```
arn:aws:iot:region:account-id:all/things
```

6. Você também pode acompanhar as violações com oListActiveViolationsAção, que permite ver quais violações foram detectadas para determinado perfil de segurança ou dispositivo de destino.

   Usar aListViolationEventsPara visualizar quais violações foram detectadas durante um período específico. Você pode filtrar esses resultados por perfil de segurança, dispositivo ou estado de verificação de alarme.
7. Você pode verificar, organizar e gerenciar seus alarmes, marcando o estado de verificação e fornecendo uma descrição desse estado de verificação, usando oPutVerificationStateOnViolationAção .
8. Se seus dispositivos violarem os comportamentos definidos com muita frequência, ou não com frequência suficiente, você deverá ajustar as definições de comportamento.
9. Para rever os perfis de segurança configurados e os dispositivos monitorados, use aListSecurityProfiles,ListSecurityProfilesForTarget, eListTargetsForSecurityProfileações.

   Usar aDescribeSecurityProfilePara obter mais detalhes sobre um perfil de segurança.
10. Para atualizar um perfil de segurança, use aUpdateSecurityProfileAção . Usar aDetachSecurityProfilePara desanexar um perfil de segurança de uma conta ou grupo de coisas de destino. Usar aDeleteSecurityProfilePara excluir completamente um perfil de segurança.

Ações de mitigação

Você pode usarAWS IoT Device DefenderPara executar ações para reduzir os problemas que foram encontrados em uma descoberta de auditoria ou Detectar um alarme.

> **Note**
>
> Ações de mitigação não serão executadas em descobertas de auditoria suprimidas. Para obter mais informações sobre supressões de descoberta de auditoria, consulteSupressões de descoberta de auditoria (p. 984).

Ações de mitigação de auditoria

AWS IoT Device Defenderfornece ações predefinidas para as diferentes verificações de auditoria. Você configura essas ações para seuConta da AWSE, em seguida, aplicá-los a um conjunto de descobertas. Essas descobertas podem ser:

- Todas as descobertas de uma auditoria. Essa opção está disponível no console do AWS IoT e por meio da AWS CLI.
- Uma lista de descobertas individuais. Essa opção só está disponível usando a AWS CLI.
- Um conjunto de descobertas filtradas de uma auditoria.

A tabela a seguir lista os tipos de verificações de auditoria e as ações de mitigação com suporte para cada um:

Mapeamento da verificação de auditoria para ações de mitigação

Verificação de auditoria	Ações de mitigação com suporte
REVOKED_CA_CERT_CHECK	PUBLISH_FINDING_TO_SNS, UPDATE_CA_CERTIFICATE
DEVICE_CERTIFICATE_SHARED_CHECK	PUBLISH_FINDING_TO_SNS, UPDATE_DEVICE_CERTIFICATE, ADD_THINGS_TO_THING_GROUP
UNAUTHENTICATED_COGNITO_ROLE_OVERLY_PERMISSIVE_CHECK	PUBLISH_FINDING_TO_SNS
AUTHENTICATED_COGNITO_ROLE_OVERLY_PERMISSIVE_CHECK	PUBLISH_FINDING_TO_SNS
IOT_POLICY_OVERLY_PERMISSIVE_CHECK	PUBLISH_FINDING_TO_SNS, REPLACE_DEFAULT_POLICY_VERSION
CA_CERT_APPROACHING_EXPIRATION_CHECK	PUBLISH_FINDING_TO_SNS, UPDATE_CA_CERTIFICATE
CONFLICTING_CLIENT_IDS_CHECK	PUBLISH_FINDING_TO_SNS
DEVICE_CERT_APPROACHING_EXPIRATION_CHECK	PUBLISH_FINDING_TO_SNS, UPDATE_DEVICE_CERTIFICATE, ADD_THINGS_TO_THING_GROUP
REVOKED_DEVICE_CERT_CHECK	PUBLISH_FINDING_TO_SNS, UPDATE_DEVICE_CERTIFICATE, ADD_THINGS_TO_THING_GROUP
LOGGING_DISABLED_CHECK	PUBLISH_FINDING_TO_SNS, ENABLE_IOT_LOGGING
DEVICE_CERTIFICATE_KEY_QUALITY_CHECK	PUBLISH_FINDING_TO_SNS, UPDATE_DEVICE_CERTIFICATE, ADD_THINGS_TO_THING_GROUP
CA_CERTIFICATE_KEY_QUALITY_CHECK	PUBLISH_FINDING_TO_SNS, UPDATE_CA_CERTIFICATE
IOT_ROLE_ALIAS_OVERLY_PERMISSIVE_CHECK	PUBLISH_FINDING_TO_SNS
IOT_ROLE_ALIAS_ALLOWS_ACCESS_TO_UNUSED_SERVICES_CHECK	PUBLISH_FINDING_TO_SNS

Todas as verificações de auditoria oferecem suporte à publicação das descobertas de auditoria no Amazon SNS para que você possa executar ações personalizadas em resposta à notificação. Cada tipo de verificação de auditoria pode oferecer suporte a ações de mitigação adicionais:

REVOKED_CA_CERT_CHECK
- Altere o estado do certificado para marcá-lo como inativo no AWS IoT.

DEVICE_CERTIFICATE_SHARED_CHECK
- Altere o estado do certificado do dispositivo para marcá-lo como inativo no AWS IoT.
- Adicione os dispositivos que usam esse certificado para um grupo de coisas.

UNAUTHENTICATED_COGNITO_ROLE_OVERLY_PERMISSIVE_CHECK
- Não há ações adicionais com suporte.

AUTHENTICATED_COGNITO_ROLE_OVERLY_PERMISSIVE_CHECK
- Não há ações adicionais com suporte.

IOT_POLICY_OVERLY_PERMISSIVE_CHECK
- Adicione uma versão de política do AWS IoT em branco para restringir as permissões.

CA_CERT_APPROACHING_EXPIRATION_CHECK
- Altere o estado do certificado para marcá-lo como inativo no AWS IoT.

CONFLICTING_CLIENT_IDS_CHECK
- Não há ações adicionais com suporte.

DEVICE_CERT_APPROACHING_EXPIRATION_CHECK
- Altere o estado do certificado do dispositivo para marcá-lo como inativo no AWS IoT.
- Adicione os dispositivos que usam esse certificado para um grupo de coisas.

DEVICE_CERTIFICATE_KEY_QUALITY_CHECK
- Altere o estado do certificado do dispositivo para marcá-lo como inativo no AWS IoT.
- Adicione os dispositivos que usam esse certificado para um grupo de coisas.

CA_CERTIFICATE_KEY_QUALITY_CHECK
- Altere o estado do certificado para marcá-lo como inativo no AWS IoT.

REVOKED_DEVICE_CERT_CHECK
- Altere o estado do certificado do dispositivo para marcá-lo como inativo no AWS IoT.
- Adicione os dispositivos que usam esse certificado para um grupo de coisas.

LOGGING_DISABLED_CHECK
- Habilitar o registro em log.

AWS IoT Device DefenderO oferece suporte aos seguintes tipos de ações de mitigação em descobertas de auditoria:

Tipo de ação	Observações
ADD_THINGS_TO_THING_GROUP	Você especifica o grupo ao qual você deseja adicionar os dispositivos. Você também pode especificar se a associação a um ou mais grupos dinâmicos deverá ser substituída se isso exceder o número máximo de grupos aos quais a coisa pode pertencer.
ENABLE_IOT_LOGGING	Você especifica o nível do registro em log e a função com permissões para registro em log. Você não pode especificar um nível de registro em log de `DISABLED`.

Tipo de ação	Observações
PUBLISH_FINDING_TO_SNS	Você especifica o tópico no qual a descoberta deve ser publicada.
REPLACE_DEFAULT_POLICY_VERSION	Você especifica o nome do modelo. Substitui a versão da política padrão por uma política padrão ou em branco. Somente um valor de `BLANK_POLICY` tem suporte atualmente.
UPDATE_CA_CERTIFICATE	Você especifica o novo estado do certificado CA. Somente um valor de `DEACTIVATE` tem suporte atualmente.
UPDATE_DEVICE_CERTIFICATE	Você especifica o novo estado para o certificado do dispositivo. Somente um valor de `DEACTIVATE` tem suporte atualmente.

Ao configurar ações padrão quando são encontrados problemas durante uma auditoria, você pode responder a esses problemas de forma consistente. Usar essas ações de mitigação definidas também ajuda a resolver os problemas mais rapidamente e com menos possibilidade de erro humano.

> **Important**
>
> Aplicar ações de mitigação que alteram os certificados, adicionar coisas a um novo grupo de coisas ou substituir a política pode ter um impacto sobre os dispositivos e aplicativos. Por exemplo, os dispositivos talvez não possam se conectar. Considere as implicações das ações de mitigação antes de aplicá-las. Talvez você precise executar outras ações para corrigir os problemas para que os dispositivos e aplicativos possam funcionar normalmente. Por exemplo, pode ser necessário fornecer certificados atualizados de dispositivos. As ações de mitigação podem ajudá-lo a limitar o risco rapidamente, mas você ainda deve executar ações corretivas para resolver os problemas subjacentes.

Algumas ações, como reativar um certificado de dispositivo, só podem ser executadas manualmente. O AWS IoT Device Defender não fornece um mecanismo para reverter automaticamente as ações de mitigação que foram aplicadas.

Detectar ações de mitigação

AWS IoT Device DefenderO oferece suporte aos seguintes tipos de ações de mitigação em Detect alarmes:

Tipo de ação	Observações
ADD_THINGS_TO_THING_GROUP	Você especifica o grupo ao qual você deseja adicionar os dispositivos. Você também pode especificar se a associação a um ou mais grupos dinâmicos deverá ser substituída se isso exceder o número máximo de grupos aos quais a coisa pode pertencer.

Como definir e gerenciar ações de mitigação

Você pode usar oAWS IoTconsole ou oAWS CLIPara definir e gerenciar ações de mitigação para oConta da AWS.

Criar ações de mitigação

Cada ação de mitigação que você define é uma combinação de um tipo de ação predefinida e dos parâmetros específicos à sua conta.

Para usar o console do AWS IoT para criar ações de mitigação.

1. Abra o console do AWS IoT.
2. No painel de navegação à esquerda, escolha Defend (Defender) e escolha Mitigation Actions (Ações de mitigação).
3. Na página Mitigation Actions (Ações de mitigação), escolha Create (Criar).

4. Na página Create a Mitigation Action (Criar uma ação de mitigação), em Action name (Nome da ação), insira um nome exclusivo para a ação de mitigação.
5. Em Action type (Tipo de ação), especifique o tipo de ação que você deseja definir.
6. Cada tipo de ação solicita um conjunto diferente de parâmetros. Insira os parâmetros para a ação. Por exemplo, se você escolher o tipo de ação Add things to thing group (Adicionar coisas ao grupo de coisas), escolha o grupo de destino e selecione ou desmarque Override dynamic groups (Substituir grupos dinâmicos).

7. Em Action execution role (Função de execução de ação), escolha a função em cujas permissões a ação é aplicada.
8. Escolha Save (Salvar) para salvar a ação de mitigação em sua conta da AWS.

Para usar a AWS CLI para criar ações de mitigação

- Usar aCreateMitigationActionPara criar a ação de mitigação. O nome exclusivo que você atribui à ação é usado quando você aplicar essa ação para auditar descobertas. Escolha um nome significativo.

Para usar o console do AWS IoT para visualizar e modificar ações de mitigação

1. Abra o console do AWS IoT.
2. No painel de navegação à esquerda, escolha Defend (Defender) e escolha Mitigation Actions (Ações de mitigação).

   OAções de mitigaçãoA página exibe uma lista de todas as ações de mitigação definidas para oConta da AWS.

3. Escolha o link do nome da ação de mitigação que você deseja alterar.
4. Faça as alterações na ação de mitigação. Como o nome da ação de mitigação é usado para identificá-la, ele não pode ser alterado.

5. SelecioneSalvarPara salvar as alterações na ação de mitigação em seuConta da AWS.

Como usar a AWS CLI para listar uma ação de mitigação

- Usar aListMitigationActionPara listar as ações de mitigação. Se você desejar alterar ou excluir uma ação de mitigação, anote o nome.

Como usar a AWS CLI para atualizar uma ação de mitigação

- Usar aUpdateMitigationActionComando para alterar a ação de mitigação.

Como usar o console do AWS IoT para excluir uma ação de mitigação

1. Abra o console do AWS IoT.
2. No painel de navegação à esquerda, escolha Defend (Defender) e escolha Mitigation Actions (Ações de mitigação).

   OAções de mitigaçãoA página exibe todas as ações de mitigação definidas para oConta da AWS.
3. Escolha as reticências (...) para a ação de mitigação que você deseja excluir e escolha Delete (Excluir).

Para usar a AWS CLI para excluir ações de mitigação

- Usar aUpdateMitigationActionComando para alterar a ação de mitigação.

Para usar o console do AWS IoT para visualizar detalhes da ação de mitigação

1. Abra o console do AWS IoT.
2. No painel de navegação à esquerda, escolha Defend (Defender) e escolha Mitigation Actions (Ações de mitigação).

   OAções de mitigaçãoA página exibe todas as ações de mitigação definidas para oConta da AWS.
3. Escolha o link do nome da ação de mitigação que você deseja alterar.
4. Na janela Are you sure you want to delete the mitigation action (Tem certeza de que deseja excluir a ação de mitigação), escolha Confirm (Confirmar).

Para usar a AWS CLI para visualizar detalhes da ação de mitigação

- Usar aDescribeMitigationActionPara visualizar detalhes da ação de mitigação.

Aplicar ações de mitigação

Depois de definir um conjunto de ações de mitigação, você pode aplicar essas ações às descobertas de uma auditoria. Ao aplicar ações, você inicia uma tarefa de ações de mitigação de auditoria. Essa tarefa pode levar algum tempo para ser concluída, dependendo do conjunto de descobertas e das ações que você aplicar a elas. Por exemplo, se você tiver um grande grupo de dispositivos cujos certificados expiraram, pode levar algum tempo para desativar todos esses certificados ou para mover esses dispositivos para um grupo de quarentena. Outras ações, como a habilitação do registro em log, podem ser concluídas rapidamente.

Você pode visualizar a lista de execuções de ações e cancelar uma execução que ainda não foi concluída. Ações já executadas como parte da execução de ações cancelada não são revertidas. Se você estiver aplicando várias ações a um conjunto de descobertas e uma dessas ações falhar, a ações subsequentes serão ignoradas para essa descoberta (mas ainda serão aplicadas a outras descobertas). O status da tarefa da descoberta é FAILED. O `taskStatus` é definido como falha se uma ou mais das ações falharem quando aplicadas às descobertas. As ações são aplicadas na ordem em que são especificadas.

Cada execução de ação aplica um conjunto de ações a um destino. Esse destino pode ser uma lista de descobertas ou todas as descobertas de uma auditoria.

O diagrama a seguir mostra como você pode definir uma tarefa de mitigação de auditoria que usa todas as descobertas de uma auditoria e aplica um conjunto de ações a essas descobertas. Uma única execução aplica uma ação a uma descoberta. A tarefa de ações de mitigação de auditoria gera um resumo da execução.

O diagrama a seguir mostra como você pode definir uma tarefa de mitigação de auditoria que usa uma lista de descobertas individuais de uma ou mais auditorias e aplica um conjunto de ações a essas descobertas. Uma única execução aplica uma ação a uma descoberta. A tarefa de ações de mitigação de auditoria gera um resumo da execução.

Você pode usar o console do AWS IoT ou a AWS CLI para aplicar ações de mitigação.

Para usar o console do AWS IoT para aplicar ações de mitigação iniciando uma execução de ações

1. Abra o console do AWS IoT.
2. No painel de navegação à esquerda, escolha Defend (Defender), escolha Audit (Auditoria) e escolha Results (Resultados).

3. Escolha o nome para a auditoria à qual você deseja aplicar ações.

4. Escolha Start Mitigation Actions (Iniciar ações de mitigação). Este botão não estará disponível se todas as suas verificações forem compatíveis.

5. Em Are you sure that you want to start mitigation action task (Tem certeza de que deseja iniciar a tarefa de ação de mitigação), o nome da tarefa usa como padrão o ID da auditoria, mas você pode alterá-lo para algo mais significativo.

6. Para cada tipo de verificação que tinha uma ou mais descobertas não compatíveis na auditoria, você pode escolher uma ou mais ações para aplicar. Somente ações válidas para o tipo de verificação são exibidas.

   Note

   Se você não tiver configurado ações para oConta da AWS, a lista de ações está vazia. Você pode escolher o link click here (clique aqui) para criar uma ou mais ações de mitigação.

7. Quando você tiver especificado todas as ações que deseja aplicar, escolha Confirm (Confirmar).

Para usar a AWS CLI para aplicar ações de mitigação iniciando uma execução de ações de mitigação de auditoria

1. Se você desejar aplicar ações a todas as descobertas da auditoria, use oListAuditTasksPara encontrar o ID da tarefa.
2. Se você desejar aplicar ações somente a descobertas selecionadas, use oListAuditFindingspara obter as IDs de descoberta.
3. Usar aListMitigationActionsComando e anote os nomes das ações de mitigação que você deseja aplicar.
4. Usar aStartAuditMitigationActionsTaskPara aplicar ações ao destino. Anote o ID da tarefa. Você pode usar o ID para verificar o estado da execução de ações, analisar os detalhes ou cancelá-la.

Para usar o console do AWS IoT para visualizar suas execuções de ações

1. Abra o console do AWS IoT.
2. No painel de navegação à esquerda, escolha Defend (Defender) e escolha Action Executions (Execuções de ações).

Uma lista de tarefas de ações que mostra quando cada uma foi iniciada e o status atual.

3. Escolha o link Name (Nome) para ver os detalhes da tarefa. Os detalhes incluem todas as ações que são aplicadas pela tarefa, seu destino e seu status.

Você pode usar os filtros Show executions for (Mostrar execuções para) para se concentrar em tipos de ações ou estados de ação.

4. Para ver detalhes da tarefa, em Executions (Execuções), escolha Show (Mostrar).

Para usar a AWS CLI para listar as tarefas iniciadas

1. Usar oListAuditMitigationActionsTasksPara visualizar as tarefas de ações de mitigação de auditoria. Você pode fornecer filtros para restringir os resultados. Se quiser exibir detalhes da tarefa, anote o ID da tarefa.
2. Usar oListAuditMitigationActionsExecutionsPara visualizar os detalhes da execução de uma tarefa de ações de mitigação de auditoria específica.
3. Usar oDescribeAuditMitigationActionsTaskPara visualizar detalhes sobre a tarefa, como os parâmetros especificados quando ela foi iniciada.

Para usar a AWS CLI para cancelar uma tarefa de ações de mitigação de auditoria em execução

1. Usar aListAuditMitigationActionsTasksPara encontrar o ID da tarefa cuja execução você deseja cancelar. Você pode fornecer filtros para restringir os resultados.
2. Usar aListDetectMitigationActionSextecutions, usando o ID da tarefa para cancelar a tarefa de ações de mitigação de auditoria. Não é possível cancelar tarefas que foram concluídas. Quando você cancelar uma tarefa, as ações restantes não serão aplicadas, mas as ações de mitigação que já foram aplicadas não serão revertidas.

Permissões

Para cada ação de mitigação que define, você deve fornecer a função usada para aplicar essa ação.

Permissões para ações de mitigação

Tipo de ação	Modelo de política de permissões	
UPDATE_DEVICE_CERTIFICATE	`{ "Version":"2012-10-17",`	

Tipo de ação	Modelo de política de permissões	

	"Statement":[
	{
	"Effect":"Allow",
	"Action":[
	"iot:UpdateCertificate"
	],
	"Resource":[
	"*"
	]
	}
	]
	}	

UPDATE_CA_CERTIFICATE	{
	"Version":"2012-10-17",
	"Statement":[
	{
	"Effect":"Allow",
	"Action":[
	"iot:UpdateCACertificate"
	],
	"Resource":[
	"*"
	]
	}
	]
	}	

ADD_THINGS_TO_THING_GROUP	{
	"Version":"2012-10-17",
	"Statement":[
	{
	"Effect":"Allow",
	"Action":[
	"iot:ListPrincipalThings",
	"iot:AddThingToThingGroup"
	],
	"Resource":[
	"*"
	]
	}
	]
	}	

Tipo de ação	Modelo de política de permissões	
REPLACE_DEFAULT_POLICY_VERSION	`{ "Version":"2012-10-17", "Statement":[ { "Effect":"Allow", "Action":[ "iot:CreatePolicyVersion" ], "Resource":[ "*" ] } ] }`	
ENABLE_IOT_LOGGING	`{ "Version":"2012-10-17", "Statement":[ { "Effect":"Allow", "Action":[ "iot:SetV2LoggingOptions" ], "Resource":[ "*" ] }, { "Effect":"Allow", "Action":[ "iam:PassRole" ], "Resource":[ "<IAM role ARN used for setting up logging>" ] } ] }`	

Tipo de ação	Modelo de política de permissões	
PUBLISH_FINDING_TO_SNS	`{ "Version":"2012-10-17", "Statement":[ { "Effect":"Allow", "Action":[ "sns:Publish" ], "Resource":[ "<The SNS topic to which the finding is published>" ] } ] }`	

Para todos os tipos de ação de mitigação, use o seguinte modelo de política de confiança:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": "iot.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "ArnLike": {
          "aws:SourceArn": "arn:aws:iot:*:111122223333::*"
        },
        "StringEquals": {
          "aws:SourceAccount": "111122223333:"
        }
      }
    }
  ]
}
```

Comandos de ação de mitigação

Você pode usar estes comandos de ação de mitigação para definir um conjunto de ações para aConta da AWSVocê pode aplicar mais tarde a um ou mais conjuntos de descobertas de auditoria. Há três categorias de comando:

- Os usados para definir e gerenciar ações.
- Os usados para iniciar e gerenciar a aplicação dessas ações em descobertas de auditoria.
- Os usados para iniciar e gerenciar a aplicação dessas ações em Detectar alarmes.

Comandos de ação de mitigação

Definir e gerenciar ações	Iniciar e gerenciar a execução de auditoria	Iniciar e gerenciar Detectar execução
CreateMitigationAction	CancelAuditMitigationActionsTask	CancelDetectMitigationActionsTask
DeleteMitigationAction	DescribeAuditMitigationActionsTask	DescribeDetectMitigationActionsTask
DescribeMitigationAction	ListAuditMitigationActionsTasks	ListDetectMitigationActionsTasks
ListMitigationActions	StartAuditMitigationActionsTask	StartDetectMitigationActionsTask
UpdateMitigationAction	ListAuditMitigationActionsExecutions	ListDetectMitigationActionSextecutions

Usar o AWS IoT Device Defender com outros serviços da AWS

O uso doAWS IoT Device Defendercom dispositivos em execuçãoAWS IoT Greengrass

AWS IoT Greengrassfornece integração pré-construída comAWS IoT Device Defenderpara monitorar os comportamentos do dispositivo de forma contínua.

- Integrar o Device DefenderAWS IoT Greengrass V1
- Integrar o Device DefenderAWS IoT Greengrass V2

O uso doAWS IoT Device Defendercom FreeRTOS e dispositivos incorporados

Para usarAWS IoT Device Defenderem um dispositivo FreeRTOS, seu dispositivo deve ter oSDK de C incorporado do FreeRTOSou oAWSBiblioteca do IoT Device Defender dainstalado. O FreeRTOS Embedded C SDK inclui oAWSBiblioteca IoT Device Defender. Para obter informações sobre como integrarAWS IoT Device Defendercom seus dispositivos FreeRTOS, veja as seguintes demonstrações:

- AWS IoT Device Defenderpara métricas padrão do FreeRTOS e demonstrações de métricas personalizadas
- Usando o agente MQTT para enviar métricas paraAWS IoT Device Defender
- Usando a biblioteca principal do MQTT para enviar métricas paraAWS IoT Device Defender

Para usarAWS IoT Device Defenderem um dispositivo incorporado sem FreeRTOS, seu dispositivo deve ter oAWSSDK de C incorporado da IoTouAWSBiblioteca do IoT Device Defender da. OAWSO SDK de C incorporado da IoT inclui oAWSBiblioteca IoT Device Defender. Para obter informações sobre como integrarAWS IoT Device Defendercom seus dispositivos incorporados, veja as seguintes demonstrações,AWS IoT Device DefenderpeloAWSDemonstrações de métricas padrão e personalizadas do IoT Embedded SDK.

Uso do AWS IoT Device Defender com a AWS IoT Device Management

Você pode usarAWS IoT Device Managementindexação de frota para indexar, pesquisar e agregar oAWS IoT Device DefenderDetectar violações. Depois que os dados de violações do Device Defender forem indexados na indexação da frota, você poderá acessar e consultar dados de violações do Device Defender de aplicativos do Fleet Hub, criar alarmes de frota com base em dados de violações para monitorar anomalias em toda a frota de dispositivos e visualizar alarmes de frota nos painéis do Fleet Hub.

> **Note**
>
> O recurso de indexação de frota para suportar indexaçãoAWS IoT Device DefenderOs dados de violações estão em versão de visualização paraAWS IoT Device ManagementE sujeita a alterações.

- Gerenciamento da indexação de frota
- Sintaxe de consulta
- Gerenciamento da indexação de frota para aplicativos Fleet Hub
- Conceitos básicos

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. Na AWS, a personificação entre serviços pode resultar no problema do confused deputy. A personificação nos 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 por meio do serviço chamado de uma forma na qual ele não deveria ter permissão para acessar. Para evitar isso, a 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.

Existem três recursosAWS IoT Device Defenderacessos de você que podem ser afetados pelo problema confuso de segurança de deptuy, executando auditorias, envio de notificações do SNS para violações de perfil de segurança e execução de ações de mitigação. Para cada uma dessas ações, os valores para`aws:SourceArn`deve ser o seguinte:

- Para recursos passados`UpdateAccountAuditConfiguration`API (atributos RoleArn e NotificationTarget RoleArn) você deve reduzir o escopo da política de recursos usando`aws:SourceArn`como`arn:arnPartition:iot:region:accountId:`.
- Para recursos passados`CreateMitigationAction`API (O atributo RoleArn) que você deve definir o escopo da política de recursos usando`aws:SourceArn`como`arn:arnPartition:iot:region:accountId:mitigationaction/mitigation`
- Para recursos passados`CreateSecurityProfile`API (o atributo alertTargets) que você deve reduzir o escopo da política de recursos usando`aws:SourceArn`como`arn:arnPartition:iot:region:accountId:securityprofile/securitypro`

A maneira mais eficaz de se proteger contra o problema confuso do deputado é usar o`aws:SourceArn`chave de contexto de condição global com o ARN completo do recurso. Se você não conhece o ARN completo do recurso ou se estiver especificando vários recursos, use

`aws:SourceArn`chave de condição de contexto global com curingas (`*`) para as partes desconhecidas do ARN. Por exemplo, `arn:aws:servicename:*:123456789012:*`.

O exemplo a seguir mostra como você pode usar `aws:SourceArn`e`aws:SourceAccount`Chaves de contexto de condição globais emAWS IoT Device DefenderPara evitar o problema confused deputy.

```
 {
"Version": "2012-10-17",
"Statement": {
  "Sid": "ConfusedDeputyPreventionExamplePolicy",
  "Effect": "Allow",
  "Principal": {
    "Service": "iot.amazonaws.com"
  },
  "Action": "sts:AssumeRole",
  "Condition": {
    "ArnLike": {
      "aws:SourceArn": "arn:aws:iot:*:123456789012::*"
    },
    "StringEquals": {
      "aws:SourceAccount": "123456789012:"
    }
  }
}
}
```

Práticas recomendadas de segurança para agentes de dispositivo

Privilégio mínimo

O processo de agente deve receber apenas as permissões mínimas necessárias para executar suas funções.

Mecanismos básicos

- O agente deve ser executado como usuário não raiz.
- O agente deve ser executado como um usuário dedicado, em seu próprio grupo.
- Usuário/grupos devem receber permissões de somente leitura para os recursos necessários para coletar e transmitir métricas.
- Exemplo: somente leitura em /proc /sys para o exemplo de agente.
- Para obter um exemplo de como configurar um processo para executar com permissões reduzidas, consulte as instruções de instalação inclusas com o exemplo de agente Python.

Existem diversos mecanismos conhecidos de Linux que podem te ajudar a restringir ou isolar ainda mais o processo de agente:

Mecanismos avançados

- CGroups
- SELinux
- Chroot
- Namespaces do Linux

Resiliência operacional

Um processo de agente deve ser resiliente a erros operacionais e exceções inesperados e não deve travar ou sair permanentemente. O código precisa processar corretamente exceções e, como precaução, ele deve ser configurado para reiniciar automaticamente em caso de encerramento inesperado (por exemplo, devido a reinícios do sistema ou exceções não detectadas).

Dependências mínimas

Um agente deve usar o menor número possível de dependências (ou seja, bibliotecas de terceiros) em sua implementação. Se o uso de uma biblioteca for justificado pela complexidade de uma tarefa (por exemplo, segurança da camada de transporte) use apenas dependências bem mantidas e estabeleça um mecanismo para mantê-las atualizadas. Se as dependências adicionais contêm funcionalidade não usada pelo agente e ativa por padrão (por exemplo, abrir portas, soquetes de domínio), desative-as no seu código ou por meio dos arquivos de configuração da biblioteca.

Isolamento do processo

Um processo de agente deve conter apenas as funcionalidades necessárias para executar a coleta e a transmissão de métricas do dispositivo. Ele não deve se aproveitar de outros processos do sistema como um contêiner ou implementar funcionalidade para outros casos de uso fora de escopo. Além disso, o processo de agente deve evitar criar canais de comunicação de entrada, como soquete de domínio e portas de serviço de rede que permitiriam que processos locais ou remotos interfiram na operação e afetem a integridade e o isolamento.

Invisibilidade

Um processo de agente não deve ser nomeado com palavras-chave, como segurança, monitoramento ou auditoria indicando sua finalidade e valor de segurança. Deve-se preferir os nomes de código genéricos ou nomes de processo aleatórios e únicos por dispositivo. O mesmo princípio deve ser seguido ao nomear o diretório em que os arquivos binários do agente residem e todos os nomes e valores dos argumentos do processo.

Mínimo de informações compartilhadas

Todos os artefatos do agente implantados em dispositivos não devem conter informações confidenciais, como credenciais privilegiadas, código de depuração e código morto, ou comentários em linha ou arquivos de documentação que revelem detalhes sobre o processamento no lado do servidor de métricas coletadas pelo agente ou outros detalhes sobre sistemas de back-end.

Transport Layer Security

Para estabelecer canais TLS seguros para transmissão de dados, um processo de agente deve impor todas as validações no lado do cliente, como cadeia de certificados e validação do nome de domínio, a nível do aplicativo, caso não esteja ativado por padrão. Além disso, um agente deve usar um armazenamento de certificados raiz que contém autoridades confiáveis e não contém os certificados que pertencem aos emissores de certificados comprometidos.

Implantação segura

Qualquer mecanismo de implantação do agente, como envio ou sincronização de código e repositórios que contém seus arquivos binários, código-fonte e quaisquer arquivos de configuração (incluindo certificados raiz confiáveis), deve ter o acesso controlado para impedir a injeção ou manipulação de código não autorizada. Se o mecanismo de implantação depende de comunicação de rede, use métodos de criptografia para proteger a integridade dos artefatos de implantação em trânsito.

Outras fontes de leitura

- Segurança em AWS IoT (p. 293)
- Noções básicas doAWS IoTModelo de segurança
- Redhat: Uma mordida de Python
- 10 common security gotchas in Python and how to avoid them
- What Is Least Privilege & Why Do You Need It?

- OWASP Embedded Security Top 10
- OWASP IoT Project

Device Advisor

Device Advisoré um recurso de teste totalmente gerenciado baseado em nuvem para validar dispositivos IoT durante o desenvolvimento de software de dispositivos. O Device Advisor fornece testes pré-criados que você pode usar para validar dispositivos IoT para conectividade confiável e segura comAWS IoT Core, antes de implantar dispositivos na produção. Os testes pré-criados do Device Advisor ajudam você a validar o software do dispositivo em relação às melhores práticas para uso deTLS,MQTT,Device Shadow, eTrabalhos de IoT. Você também pode baixar relatórios de qualificação assinados para enviar aoAWSRede de parceiros para qualificar seu dispositivo para oAWSCatálogo de dispositivos para parceirossem a necessidade de enviar seu dispositivo e esperar que ele seja testado.

> **Note**
>
> O Device Advisor é compatível com as regiões us-east-1, ap-northeast-1 e eu-west-1. O Device Advisor suporta MQTT com certificados de cliente X509.

Este capítulo contém as seguintes seções:

- Configuração (p. 1067)
- Primeiros passos com o Device Advisor no console (p. 1071)
- Fluxo de trabalho do Device (p. 1077)
- Fluxo de trabalho detalhado do console Device (p. 1081)
- Casos de testes do Device (p. 1091)

Qualquer dispositivo que tenha sido construído para se conectar aAWS IoT Corepode tirar proveito do Device Advisor. Você pode acessar o Device Advisor doAWS IoTconsole, ou usando oAWS CLIou SDK. Quando você estiver pronto para testar seu dispositivo, registre-o comAWS IoT Coree configure o software do dispositivo com o endpoint Device Advisor. Em seguida, escolha os testes pré-criados, configure-os, execute os testes em seu dispositivo e obtenha os resultados do teste junto com logs detalhados ou um relatório de qualificação.

O Device Advisor é um endpoint de teste naAWSnuvem. Você pode testar seus dispositivos configurando-os para se conectar ao endpoint de teste fornecido pelo Device Advisor. Depois que um dispositivo é configurado para se conectar ao endpoint de teste, você pode visitar o console do Device Advisor ou usar oAWSO SDK para escolher os testes que você deseja executar em seus dispositivos. O Device Advisor gerencia todo o ciclo de vida de um teste, incluindo o provisionamento de recursos, o agendamento do processo de teste, o gerenciamento da máquina de estado, a gravação do comportamento do dispositivo, o registro dos resultados e o fornecimento dos resultados finais na forma de um relatório de teste.

Configuração

Antes de usar o Device Advisor pela primeira vez, conclua as seguintes tarefas.

Pré-requisitos

- Criar uma coisa da IoT (p. 1068)
- Crie uma função do IAM para ser usada como função de dispositivo (p. 1068)
- Criar uma política gerenciada personalizada para um usuário do IAM usar o Device Advisor (p. 1069)

- Criar um usuário do IAM para usar o Device Advisor (p. 1070)
- Configurar o dispositivo (p. 1070)

Criar uma coisa da IoT

Primeiro você precisará criar uma coisa e anexar um certificado à coisa. Use o seguinte tutorial para criar uma coisa: Criar um objeto de coisa.

Crie uma função do IAM para ser usada como função de dispositivo

> Note
>
> Você pode criar rapidamente a função do dispositivo usando o console do Device Advisor. ConsulteConceitos básicos do Device Advisor no consolepara obter as etapas para configurar a função do dispositivo usando o console do Device Advisor.

1. Acesse oAWSConsole do IAMe faça login na conta que você usa para testes do Device Advisor.
2. No painel de navegação à esquerda, escolhaPolíticas.
3. Escolha Create policy (Criar política).
4. UNDERCriar política, faça o seguinte:
   1. para oServiço, escolhaIoT.
   2. UNDERAções, selecione ações com base na política anexada à coisa de IoT ou certificado criado na seção anterior (recomendado) ou pesquise as seguintes ações noFiltrocaixa de ação e selecione-os.
      - Conecte-se
      - Publicar
      - Inscrever-se
      - Receber
   3. UNDERRecursos, ou melhores práticas de segurança, recomendamos que você restrinja ocliente,tópico, efiltro de tópicosRecursos do usando as seguintes etapas:
      a. SelecioneEspecificar ARN de recurso cliente para a ação Connect.
         i. Escolha Add ARN (Adicionar ARN).
         ii. Especifique oregião,accountId, eclientIdno editor de ARN visual ou especifique manualmente os nomes de recursos da Amazon (ARNs) dos tópicos de IoT que você deseja usar para executar casos de teste. OclientIdÉ o MQTTclientIdseu dispositivo usa para interagir com o Device Advisor.
         iii. Escolha Add (Adicionar).
      b. SelecioneEspecifique o ARN de recurso de tópico para a ação Receber e mais 1.
         i. Escolha Add ARN (Adicionar ARN).
         ii. Especifique oregião,accountId, eNome do tópicono editor de ARN visual ou especifique manualmente os ARNs dos tópicos de IoT que você deseja usar para executar casos de teste. ONome do tópicoÉ o MQTTtópicoseu dispositivo usa para publicar mensagens no.
         iii. Escolha Add (Adicionar).
      c. SelecioneEspecificar ARN de recurso de filtro de tópicos para a ação Assinar.
         i. Escolha Add ARN (Adicionar ARN).
         ii. Especifique oregião,accountId, eNome do tópicono editor de ARN visual ou especifique manualmente os ARNs dos tópicos de IoT que você deseja usar para executar casos de teste. ONome do tópicoÉ o MQTTtópicoseu dispositivo usa para se inscrever.

iii. Escolha Add (Adicionar).

5. Escolha Review policy (Revisar política).
6. UNDERPolítica de revisão do, insira umName (Nome).
7. Escolha Create policy (Criar política).
8. No painel de navegação à esquerda, EscolhaFunções do.
9. Selecione Create Role (Criar função).
10. UNDEROu selecione um serviço para visualizar seus casos de uso, escolhaIoT.
11. Em Select your use case (Selecione seu caso de uso), escolha IoT.
12. Selecione Next (Próximo): Permissions
13. (Opcional) EmDefinir limite de permissões, EscolhaUse um limite de permissões para controlar o número máximo de permissões de funçõesE escolha a política que você acabou de criar.
14. Selecione Next (Próximo): Tags.
15. Selecione Next (Próximo): Análise.
16. Digite umNome da funçãoe umDescrição de função.
17. Selecione Create role (Criar função).
18. Navegue até a função que você criou.
19. NoPermissõesGuia, selecioneAnexar políticasEm seguida, escolha a política que você criou na Etapa 4
20. Escolha Attach policy (Anexar política).
21. SelecioneRelações de confiançaguia e escolhaEditar relação de confiança.
22. Insira esta política:

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

23. Escolha Update Trust Policy.

Criar uma política gerenciada personalizada para um usuário do IAM usar o Device Advisor

1. Navegue até o console do IAM emhttps://console.aws.amazon.com/iam/. Se solicitado, insira oAWSCredenciais do para fazer login.
2. No painel de navegação à esquerda, escolha Policies (Políticas).
3. SelecioneCriar políticae, depois, escolha oJSONGuia.
4. Adicione as permissões necessárias para usar o Device Advisor. O documento de política pode ser encontrado no tópicoPráticas recomendadas de segurança.
5. Escolha Review Policy (Revisar política).
6. Preencha os campos Name (Nome) e Description (Descrição).

7. Escolha Create Policy (Criar política).

Criar um usuário do IAM para usar o Device Advisor

> **Note**
>
> Recomendamos que você crie um usuário do IAM para usar ao executar testes do Device Advisor de Dispositivos. Usar um usuário administrador do IAM para executar testes do Device Advisor, embora permitido, não é recomendado.

1. Navegue até o console do IAM emhttps://console.aws.amazon.com/iam/Se solicitado, insira oAWSCredenciais do para fazer login.
2. No painel de navegação à esquerda, EscolhaUsuários do.
3. Escolha Add User (Adicionar usuário).
4. Digite umNome de usuário.
5. SelectAcesso programático.
6. Selecione Next (Próximo): Permissions
7. Escolha Attach existing policies directly (Anexar políticas existentes diretamente).
8. Digite o nome da política gerenciada personalizada que você criou na caixa de pesquisa e, em seguida, marque a caixa de seleção paraNome da política.
9. Selecione Next (Próximo): Tags.
10. Selecione Next (Próximo): Análise.
11. Escolha Create user (Criar usuário).
12. Escolha Close (Fechar).

O Device Advisor requer acesso aoAWSrecursos (coisas, certificados e endpoints) em seu nome. O usuário do IAM deve ter as permissões necessárias. O Device Advisor também publicará logs na Amazon CloudWatch Se você anexar a política de permissões necessária ao usuário do IAM.

Configurar o dispositivo

O Device Advisor usa a extensão TLS do Server Name Indication (SNI — Indicação de Nome de Servidor) para aplicar configurações TLS. Os dispositivos devem usar essa extensão ao conectar e transmitir um nome de servidor idêntico ao endpoint de teste do Device Advisor.

O Device Advisor permite a conexão TLS quando um teste está no`Running`state e nega a conexão TLS antes e depois de cada execução de teste. Por esse motivo, também recomendamos usar o mecanismo de repetição de conexão do dispositivo para ter uma experiência de teste totalmente automatizada com o Device Advisor. Se você executar um conjunto de testes com mais de um caso de teste (por exemplo, conexão TLS, MQTT connect e MQTT publish), recomendamos que você tenha um mecanismo criado para que seu dispositivo tente se conectar ao nosso endpoint de teste a cada cinco segundos. Em seguida, você pode executar vários casos de teste, em sequência, de forma automatizada.

> **Note**
>
> Para tornar o software do seu dispositivo pronto para testes, recomendamos que você tenha um SDK que possa se conectar aoAWS IoT Coree atualize o SDK com o endpoint de teste do Device Advisor fornecido para sua conta.

O Device Advisor oferece suporte a dois tipos de endpoints: Endpoints em nível de conta e endpoints no nível do dispositivo. Selecione o endpoint que seja mais adequado ao caso de uso. Para executar simultaneamente vários conjuntos de testes usando dispositivos diferentes, use um endpoint no nível do dispositivo. Execute o comando a seguir para obter o endpoint no nível do dispositivo:

```
aws iotdeviceadvisor get-endpoint --thing-arn your-thing-arn
```

ou

```
aws iotdeviceadvisor get-endpoint --certificate-arn your-certificate-arn
```

Para executar um conjunto de testes por vez, escolha um endpoint no nível da conta. Execute o comando a seguir para obter o endpoint no nível da conta:

```
aws iotdeviceadvisor get-endpoint
```

Primeiros passos com o Device Advisor no console

Este tutorial ajuda você a começar rapidamente a usar o Device Advisor no console. O Device Advisor oferece recursos como testes necessários e relatórios de qualificação assinados para qualificar e listar dispositivos naAWSCatálogo de dispositivos para parceirosconforme detalhado noAWS IoT CorePrograma de qualificação do programa.

Para obter mais informações sobre o uso do Device Advisor, consulteFluxo de trabalho do Device (p. 1077)eFluxo de trabalho detalhado do console Device (p. 1081).

Para concluir este tutorial, siga as etapas descritas emConfiguração (p. 1067).

> **Note**
>
> O Device Advisor é compatível com as regiões us-east-1, ap-northeast-1 e eu-west-1.

Começar a usar

1. NoAWS IoTconsoleNo painel de Navegação, expandaTeste, e depoisDevice AdvisorEscolha e, depois, escolhaDemonstração do início.

2. OConceitos básicos do Device Advisora página fornece uma visão geral das etapas necessárias para criar um conjunto de testes e executar testes em seu dispositivo. Você também pode encontrar o endpoint de teste do Device Advisor de dispositivo para sua conta. Você deve configurar o firmware ou o software no dispositivo que você usará para testes para se conectar a esse endpoint de teste.

Para concluir este tutorial, você precisaCriar uma coisa e certificado.

Depois de revisar as informações, escolhaPróximo.

3. DentroEtapa 1, selecione umAWS IoTcoisa ou certificado para testar usando o Device Advisor. Se você não tiver nenhuma coisa ou certificado, consulteConfigurar o.

   NoTestar endpoint, selecione o endpoint que melhor se adapta ao nosso caso de uso. Se você planeja executar vários conjuntos de testes simultaneamente usando o mesmoAWSConta, selecioneEndpoint no nível do dispositivo. Caso contrário, para executar um conjunto de testes por vez, selecionePonto de extremidade no nível da conta.

   Configure seu dispositivo de teste com o endpoint de teste do Device Advisor selecionado.

   Escolha Next (Próximo).

4. DentroEtapa 2, você cria e configura um conjunto de testes personalizado. Um conjunto de testes personalizado deve ter pelo menos um grupo de teste, e cada grupo de teste deve ter pelo menos um caso de teste. Adicionamos oMQTT Connectcaso de teste para você começar.

   SelecionePropriedades do conjunto de testes. Você deve fornecer as propriedades do conjunto de testes ao criar seu conjunto de testes.

Você pode configurar as seguintes propriedades no nível do suite-level:

- Nome do conjunto de testes: Insira um nome para o conjunto de testes.
- Timeout (Tempo limite)(opcional): O tempo limite em segundos para cada caso de teste no conjunto de testes atual. Se você não especificar um valor de tempo limite, o valor padrão será usado.
- Tags(opcional): Adicione tags ao conjunto de testes que você criará.

Quando você terminar, hooseAtualizar propriedades.

5. (Opcional) Você pode atualizar a configuração do grupo de suítes de testes escolhendoEditeAo lado do nome do grupo de teste.

   - Name (Nome): Insira um nome personalizado para o grupo de conjuntos de testes.
   - Timeout (Tempo limite)(opcional): O tempo limite em segundos para cada caso de teste no conjunto de testes atual. Se você não especificar um valor de tempo limite, o valor padrão será usado.

Escolha Done (Concluído).

6. (Opcional) Você pode atualizar a configuração do caso de teste escolhendoEditeao lado do nome do caso de teste.

   - Name (Nome): Insira um nome personalizado para o grupo de conjuntos de testes.
   - Timeout (Tempo limite)(opcional): O tempo limite em segundos para o caso de teste selecionado. Se você não especificar um valor de tempo limite, o valor padrão será usado.

Escolha Done (Concluído).

7. (Opcional) Para adicionar mais grupos de teste ao conjunto de testes, escolhaAdicionar grupo de testese ollow as instruções na Etapa 5..
8. (Opcional) Para adicionar mais casos de teste, arraste os casos de teste exibidos emCasos de testeEm qualquer um dos seus grupos de testes.

9. Grupos de teste e casos de teste podem ser reordenados selecionando e arrastando os casos de teste listados. O Device Advisor executa testes na ordem em que seus casos de teste estão listados.

   Depois de configurar sua suíte de testes, escolhaPróximo.

10. DentroEtapa 3você pode configurar uma função de dispositivo que o Device Advisor usará para executarAWS IoTAções MQTT em nome do seu dispositivo de teste. Se você selecionou oMQTT Connectsomente caso de teste, oConecte-sea ação será selecionada automaticamente, pois essa permissão é necessária na função do dispositivo para executar este conjunto de testes. Se você selecionou outros casos de teste, as ações correspondentes serão selecionadas.

    Forneça os valores de recursos para cada uma das ações selecionadas. Por exemplo, para oConecte-se, forneça o ID do cliente com o qual o dispositivo se conectará ao endpoint do Device Advisor. Você pode fornecer vários valores usando vírgulas para separar os valores e fornecer valores de prefixo usando um caractere curinga (*). Por exemplo, para fornecer permissão para publicar em qualquer tópico que comece com`MyTopic`, você pode fornecer"`MyTopic*`" como o valor do recurso.

    Se você já criou uma função de dispositivo anteriormente a partir deConfigurar oE você gostaria de usar essa função, escolhaSelecione uma função existentee escolha a função do dispositivo emSelecionar função.

Configure a função do dispositivo usando uma das duas opções fornecidas e escolhaPróximo.

11. Etapa 4mostra uma visão geral do dispositivo de teste selecionado, endpoint de teste, conjunto de testes e função de dispositivo de teste que você configurou. Se você deseja fazer alterações em uma seção, escolha aEditebotão acima da seção que você deseja editar.

    > **Note**
    >
    > Para obter melhores resultados, você pode conectar o dispositivo de teste selecionado ao endpoint de teste do Device Advisor antes de iniciar a execução do conjunto de testes. Recomendamos que você tenha um mecanismo criado para que seu dispositivo tente se conectar ao nosso endpoint de teste a cada cinco segundos por um a dois minutos.

    Para criar o conjunto de testes e executar os testes selecionados em seu dispositivo, escolhaExecução do.

12. No painel de Navegação, expandaTeste,Device AdvisorEscolha e, depois, escolhaTestes e resultadosPara visualizar os detalhes da execução e logs. Selecione a execução do conjunto de testes que você começou para exibir os detalhes e os registros da execução.

13. Para acessar o CloudWatch logs para a execução da suíte:
    - SelecioneLog do conjunto de testesPara visualizar o CloudWatch Os logs da execução do conjunto de testes.
    - SelecioneRegistro de casos de testepara qualquer caso de teste para visualizar o caso de teste específico CloudWatch logs.
14. Com base nos resultados do teste,solucionar problemasSeu dispositivo até que todos os testes sejam aprovados.

Fluxo de trabalho do Device

Este tutorial fornece instruções sobre como criar um conjunto de testes personalizado e executar testes no dispositivo que você deseja testar no console. Após a conclusão dos testes, você pode exibir os resultados do teste e os logs detalhados.

Tutoriais

- Pré-requisitos (p. 1077)
- Criar uma definição de suíte de testes (p. 1077)
- Obtenha uma definição de suíte de testes (p. 1079)
- Obter um endpoint de teste (p. 1079)
- Iniciar uma execução de suíte de testes (p. 1080)
- Faça uma execução de suíte de testes (p. 1080)
- Pare a execução de um conjunto de testes (p. 1080)
- Obtenha um relatório de qualificação para uma execução bem-sucedida do conjunto de testes de qualificação (p. 1081)

Pré-requisitos

Para concluir este tutorial, conclua as etapas descritas emConfiguração (p. 1067).

Criar uma definição de suíte de testes

Primeiro,instalar umAWSSDK.

Sintaxe de `rootGroup`

Um grupo raiz é uma string JSON que especifica quais casos de teste estão incluídos no conjunto de testes, bem como todas as configurações necessárias para esses casos de teste. Use o grupo raiz para

estruturar e ordenar seu conjunto de testes da maneira que você quiser. A hierarquia de um conjunto de testes é:

```
test suite # test group(s) # test case(s)
```

Um conjunto de testes deve ter pelo menos um grupo de teste, e cada grupo de teste deve ter pelo menos um caso de teste. O Device Advisor executa testes na ordem em que você define os grupos de teste e os casos de teste.

Cada grupo raiz segue essa estrutura básica:

```
{
    "configuration": {  // for all tests in the test suite
        "": ""
    }
    "tests": [{
        "name": ""
        "configuration": {  // for all sub-groups in this test group
            "": ""
        },
        "tests": [{
            "name": ""
            "configuration": {  // for all test cases in this test group
                "": ""
            },
            "test": {
                "id": ""
                "version": ""
            }
        }]
    }]
}
```

Um bloco que contém um`"name"`,`"configuration"`, e`"tests"`é referido como uma “definição de grupo”. Um bloco que contém um`"name"`,`"configuration"`, e`"test"`é referido como uma “definição de caso de teste”. EACH`"test"`bloco que contém um`"id"`e`"version"`é referido como um “caso de teste”.

Para obter informações sobre como preencher o`"id"`e`"version"`campos para cada caso de teste (`"test"`bloco), consulteCasos de testes do Device (p. 1091). Essa seção também contém informações sobre o disponível`"configuration"`Configurações do .

O bloco a seguir é um exemplo de uma configuração de grupo raiz que especifica os casos de teste “MQTT Connect Happy Case” e “MQTT Connect Exponential Backoff Retries”, juntamente com descrições dos campos de configuração.

```
{
    "configuration": {},  // Suite-level configuration
    "tests": [            // Group definitions should be provided here
      {
        "name": "My_MQTT_Connect_Group",  // Group definition name
        "configuration": {}               // Group definition-level configuration,
        "tests": [                        // Test case definitions should be provided here
        {
            "name": "My_MQTT_Connect_Happy_Case",  // Test case definition name
            "configuration": {
                "EXECUTION_TIMEOUT": 300        // Test case definition-level
 configuration, in seconds
            },
            "test": {
                "id": "MQTT_Connect",              // test case id
                "version": "0.0.0"                 // test case version
            }
```

```
        },
        {
            "name": "My_MQTT_Connect_Jitter_Backoff_Retries",  // Test case definition name
            "configuration": {
                "EXECUTION_TIMEOUT": 600                 // Test case definition-level
 configuration,  in seconds
            },
            "test": {
                "id": "MQTT_Connect_Jitter_Backoff_Retries",  // test case id
                "version": "0.0.0"                                // test case version
            }
        }]
    }]
}
```

Você deve fornecer a configuração do grupo raiz ao criar a definição do conjunto de testes. Salvar o`suiteDefinitionId`que é retornado no objeto de resposta. Esse ID é usado para recuperar as informações de definição do conjunto de testes e para executar o conjunto de testes.

Aqui está um exemplo do Java SDK:

```
response = iotDeviceAdvisorClient.createSuiteDefinition(
        CreateSuiteDefinitionRequest.builder()
            .suiteDefinitionConfiguration(SuiteDefinitionConfiguration.builder()
                .suiteDefinitionName("your-suite-definition-name")
                .devices(
                    DeviceUnderTest.builder()
                        .thingArn("your-test-device-thing-arn")
                        .certificateArn("your-test-device-certificate-arn")
                        .build()
                )
                .rootGroup("your-root-group-configuration")
                .devicePermissionRoleArn("your-device-permission-role-arn")
                .build()
            )
            .build()
)
```

Obtenha uma definição de suíte de testes

Depois de iniciar uma execução de suíte de testes, você pode verificar seu progresso e seus resultados com o`GetSuiteRun`API.

Exemplo do SDK:

```
// Using the SDK, call the GetSuiteRun API.

response = iotDeviceAdvisorClient.GetSuiteRun(
GetSuiteRunRequest.builder()
    .suiteDefinitionId("your-suite-definition-id")
    .suiteRunId("your-suite-run-id")
.build())
```

Obter um endpoint de teste

Você pode usar o`GetTestEndpoint`API para obter o endpoint de teste usado pelo dispositivo. Ao escolher o endpoint, selecione o endpoint que melhor se adapta à situação. Para executar simultaneamente vários conjuntos de testes, use o endpoint no nível do dispositivo fornecendo um`thing ARN`ou um`certificate ARN`. Para executar um único conjunto de testes, escolha o endpoint no nível da conta não fornecendo argumentos.

Exemplo do SDK:

```
response = iotDeviceAdvisorClient.getEndpoint(GetEndpointRequest.builder()
.certificateArn("your-test-device-certificate-arn")
.thingArn("your-test-device-thing-arn")
.build())
```

Iniciar uma execução de suíte de testes

Depois de criar com êxito uma definição de conjunto de testes e configurar o dispositivo de teste para se conectar ao endpoint de teste do Device Advisor, execute o conjunto de testes com o`StartSuiteRun`API. Use um dos dois`certificateArn`ou`thingArn`para executar o conjunto de testes. Se ambos estiverem configurados, o certificado será usado se pertencer à coisa.

para o`.parallelRun()`, use`true`se você usar endpoint no nível do dispositivo para executar vários conjuntos de teste em parallel usando umAWSconta.

Exemplo do SDK:

```
response = iotDeviceAdvisorClient.startSuiteRun(StartSuiteRunRequest.builder()
.suiteDefinitionId("your-suite-definition-id")
.suiteRunConfiguration(SuiteRunConfiguration.builder()
    .primaryDevice(DeviceUnderTest.builder()
        .certificateArn("your-test-device-certificate-arn")
        .thingArn("your-test-device-thing-arn")
        .build())
    .parallelRun(true | false)
    .build())
.build())
```

Salvar o`suiteRunId`O é retornado na resposta. Você usará isso para recuperar os resultados desta execução do conjunto de testes.

Faça uma execução de suíte de testes

Depois de criar sua definição de suíte de testes, você receberá o`suiteDefinitionId`No objeto de resposta do`CreateSuiteDefinition`API.

Você pode ver que há novos`id`campos dentro de cada uma das definições de caso de teste e grupo no grupo raiz retornado. Isso é esperado; você pode usar esses IDs para executar um subconjunto da definição do conjunto de testes.

Exemplo do SDK do Java:

```
response = iotDeviceAdvisorClient.GetSuiteDefinition(
    GetSuiteDefinitionRequest.builder()
        .suiteDefinitionId("your-suite-definition-id")
        .build()
)
```

Pare a execução de um conjunto de testes

Para interromper a execução de uma suíte de testes que ainda está em andamento, você pode chamar o`StopSuiteRun`API. Depois de ligar para o`StopSuiteRun`API, o serviço iniciará o processo de limpeza. Enquanto o serviço está executando o processo de limpeza, o status de execução do conjunto de testes é atualizado para`Stopping`. O processo de limpeza levará vários minutos e, quando o processo for

concluído, o status de execução do conjunto de testes será atualizado para`Stopped`. Depois que uma execução de teste parar completamente, você poderá iniciar outra execução do conjunto de testes. Você pode verificar periodicamente o status de execução da suíte usando o`GetSuiteRun`API como mostrado na seção anterior.

Exemplo do SDK:

```
// Using the SDK, call the StopSuiteRun API.

response = iotDeviceAdvisorClient.StopSuiteRun(
StopSuiteRun.builder()
    .suiteDefinitionId("your-suite-definition-id")
    .suiteRunId("your-suite-run-id")
.build())
```

Obtenha um relatório de qualificação para uma execução bem-sucedida do conjunto de testes de qualificação

Se você executar um conjunto de testes de qualificação concluído, poderá recuperar um relatório de qualificação usando o`GetSuiteRunReport`API. Você pode usar esse relatório de qualificação para qualificar seu dispositivo com oAWS IoT CorePrograma de qualificação do. Para determinar se o conjunto de testes é um conjunto de testes de qualificação, verifique se o`intendedForQualification`O parâmetro está definido como`true`. Depois de ligar para o`GetSuiteRunReport`API, o URL de download retornado está disponível para download por 90 segundos. Se decorrerem mais de 90 segundos da hora anterior, você chamou o`GetSuiteRunReport`API, chame a API novamente para recuperar um URL válido.

Exemplo do SDK:

```
// Using the SDK, call the getSuiteRunReport API.

response = iotDeviceAdvisorClient.getSuiteRunReport(
    GetSuiteRunReportRequest.builder()
        .suiteDefinitionId("your-suite-definition-id")
        .suiteRunId("your-suite-run-id")
        .build()
)
```

Fluxo de trabalho detalhado do console Device

Neste tutorial, você criará um conjunto de testes personalizado e executará testes no dispositivo que deseja testar no console. Após a conclusão dos testes, você pode exibir os resultados do teste e os logs detalhados.

Tutoriais

- Pré-requisitos (p. 1082)
- Criar uma definição de suíte de testes (p. 1082)
- Iniciar uma execução de suíte de testes (p. 1086)
- Interromper a execução da suíte de testes (opcional) (p. 1088)
- Exibir detalhes e registros de execução da suíte de testes (p. 1089)
- Baixar umAWS IoTRelatório de qualificação do relatório (p. 1090)

Pré-requisitos

Para concluir este tutorial, você precisaCriar uma coisa e certificado.

Criar uma definição de suíte de testes

1. NoAWS IoTconsoleNo painel de Navegação, expandaTeste,Device AdvisorEscolha e, depois, escolhaConjunto de testes do.

   SelecioneCriar conjunto de testes do.

2. Selecione um`Use the AWS Qualification test suite`ou`Create a new test suite`.

   Select`Use the AWS Qualification test suite`para qualificar e listar seu dispositivo para oAWSCatálogo de dispositivos de parceiros. Ao escolher essa opção, casos de teste necessários para a qualificação do seu dispositivo para oAWS IoT Coreprograma de qualificação são pré-selecionados. Grupos de teste e casos de teste não podem ser adicionados ou removidos. Você ainda precisará configurar as propriedades do conjunto de testes.

   Select`Create a new test suite`para criar e configurar um conjunto de testes personalizado. Recomendamos começar com essa opção para testes iniciais e solução de problemas. Um conjunto de testes personalizado deve ter pelo menos um grupo de teste, e cada grupo de teste deve ter pelo menos um caso de teste. Para fins deste tutorial, selecionaremos essa opção e escolhemosPróximo.

3. SelecionePropriedades do conjunto de testes. Você deve criar as propriedades do conjunto de testes ao criar sua suíte de testes.

UNDERPropriedades do conjunto de testes, preencha o seguinte:

- Nome do conjunto de testes: Você pode criar a suíte com um nome personalizado.
- Timeout (Tempo limite)(opcional): O tempo limite em segundos para cada caso de teste no conjunto de testes atual. Se você não especificar um valor de tempo limite, o valor padrão será usado.
- Tags(opcional): Adicione tags à suíte de testes.

Quando terminar, escolhaAtualizar propriedades.

4. Para modificar a configuração do nível de grupo, em`Test group 1`, escolhaEdite. Em seguida, digite umName (Nome)para dar ao grupo um nome personalizado.

   Se preferir, também é possível inserir umTimeout (Tempo limite)valor em segundos sob o grupo de teste selecionado. Se você não especificar um valor de tempo limite, o valor padrão será usado.

   Escolha Done (Concluído).

5. Arraste um dos casos de teste disponíveis deCasos de testepara o grupo de testes.

6. Para modificar a configuração do nível do caso de teste para o caso de teste que você adicionou ao grupo de teste, escolhaEdite. Em seguida, digite umName (Nome)para dar ao grupo um nome personalizado.

   Se preferir, também é possível inserir umTimeout (Tempo limite)valor em segundos sob o grupo de teste selecionado. Se você não especificar um valor de tempo limite, o valor padrão será usado.

   Escolha Done (Concluído).

   > **Note**
   >
   > Para adicionar mais grupos de teste ao conjunto de testes, escolhaAdicionar grupo de testes. Siga as etapas anteriores para criar e configurar mais grupos de teste ou adicionar mais casos de teste a um ou mais grupos de teste. Grupos de teste e casos de teste podem ser reordenados escolhendo e arrastando um caso de teste para a posição desejada. O Device Advisor executa testes na ordem em que você define os grupos de teste e os casos de teste.

7. Escolha Next (Próximo).
8. DentroEtapa 3, configure uma função de dispositivo que o Device Advisor usará para executarAWS IoTAções MQTT em nome do seu dispositivo de teste.

   Se você selecionouMQTT Connectcaso de teste somente emEtapa 2, oConecte-sea ação será verificada automaticamente, pois essa permissão é necessária na função do dispositivo para executar este conjunto de testes. Se você selecionou outros casos de teste, as ações necessárias correspondentes serão verificadas. Certifique-se de que os valores de recursos para cada uma das ações sejam fornecidos. Por exemplo, para oConecte-se, forneça o ID do cliente com o qual o dispositivo se conectará ao endpoint do Device Advisor. Você pode fornecer vários valores usando vírgulas para separar os valores e fornecer valores de prefixo usando um caractere curinga (*)

também. Por exemplo, para fornecer permissão para publicar em qualquer tópico que comece com`MyTopic`, você pode fornecer"`MyTopic*`" como o valor do recurso.

Se você já criou uma função de dispositivo anteriormente e gostaria de usar essa função, selecioneSelecione uma função existentee escolha a função do dispositivo emSelecionar função.

Configure a função do dispositivo usando uma das duas opções fornecidas e escolhaPróximo.

9. DentroEtapa 4, certifique-se de que a configuração fornecida em cada uma das etapas seja precisa. Para editar a configuração fornecida para uma etapa específica, escolhaEditepara a etapa correspondente.

   Depois de verificar a configuração, escolhaCriar conjunto de testes do.

   A suíte de testes deve ser criada com sucesso e você será redirecionado para oConjunto de testes doPágina em que você pode exibir todo o conjunto de testes que foi criado.

   Se a criação do conjunto de testes falhar, verifique se o conjunto de testes, os grupos de teste, os casos de teste e a função do dispositivo foram configurados de acordo com as instruções anteriores.

Iniciar uma execução de suíte de testes

1. NoAWS IoTconsoleNo painel de Navegação, expandaTeste,Device AdvisorEscolha e, depois, escolhaConjunto de testes do.
2. Escolha o conjunto de testes para o qual você gostaria de ver os detalhes da suíte de testes.

A página de detalhes do conjunto de testes exibe todas as informações relacionadas ao conjunto de testes.

3. SelecioneAções, entãoExecute o conjunto de testes.

4. UNDERConfiguração da execução, você precisará selecionar umAWS IoTcoisa ou certificado para testar usando o Device Advisor. Se você não tiver nenhuma coisa ou certificado, primeirocriarAWS IoT Corerecursos (p. 1067).

   DentroTestar endpoint, selecione o endpoint que melhor se adapta ao seu caso. Se você planeja executar vários conjuntos de testes simultaneamente usando o mesmoAWSconta no future, selecioneEndpoint no nível do dispositivo. Caso contrário, se você planeja executar apenas um conjunto de testes por vez, selecionePonto de extremidade no nível da conta.

   Configure seu dispositivo de teste com o endpoint de teste do Device Advisor selecionado.

   Depois de selecionar uma coisa ou certificado e escolher um endpoint do Device Advisor, escolhaExecute o teste.

5. SelecioneAcesse resultadosno banner superior para visualizar os detalhes da execução de teste.

Interromper a execução da suíte de testes (opcional)

1. NoAWS IoTconsoleNo painel de Navegação, expandaTeste,Device AdvisorEscolha e, depois, escolhaTestes e resultados.
2. Selecione a suíte de testes em andamento que você deseja interromper.

3. SelecioneAções, entãoConjunto de testes de parada.

4. O processo de limpeza levará alguns minutos para ser concluído. Enquanto o processo de limpeza for executado, o status da execução do teste será`STOPPING`. Aguarde até que o processo de limpeza seja concluído e que o status do conjunto de testes mude para o`STOPPED`status antes de iniciar uma nova execução de suíte.

Exibir detalhes e registros de execução da suíte de testes

1. NoAWS IoTconsoleNo painel de Navegação, expandaTeste,Device AdvisorEscolha e, depois, escolhaTestes e resultados.

   Esta página exibe:

   - Número de coisas de IoT
   - Número de certificados de IoT
   - Número de conjuntos de testes em execução no momento
   - Todas as execuções do conjunto de testes que foram criadas
2. Escolha o conjunto de testes para o qual você gostaria de ver os detalhes e os registros da execução.

A página de resumo da execução exibe o status da execução atual do conjunto de testes. Esta página é atualizada automaticamente a cada 10 segundos. Recomendamos que você tenha um mecanismo criado para que seu dispositivo tente se conectar ao nosso endpoint de teste a cada cinco segundos por um a dois minutos. Em seguida, você pode executar vários casos de teste em sequência de forma automatizada.

3. Para acessar o CloudWatch logs para a execução do conjunto de testes, escolhaRegistro do conjunto de testes.

   Para acessar o CloudWatch logs para qualquer caso de teste, escolhaRegistro de casos de teste.

4. Com base nos resultados do teste,solucionar problemasSeu dispositivo até que todos os testes sejam aprovados.

Baixar umAWS IoTRelatório de qualificação do relatório

Se você escolheu oUsar aAWS IoTConjunto de testes de qualificação doao criar um conjunto de testes e conseguir executar um conjunto de testes de qualificação, você pode baixar um relatório de qualificação escolhendoFazer download do relatório de qualificaçãona página de resumo da execução do teste.

Casos de testes do Device

O Device Advisor fornece testes pré-criados em cinco categorias.

- TLS (p. 1091)
- MQTT (p. 1095)
- Shadow (p. 1103)
- Execução de trabalho (p. 1104)
- Políticas e permissões (p. 1105)

Casos de teste do Device Advisor para se qualificar para oAWSPrograma de qualificação do dispositivo.

Seu dispositivo precisa passar nos seguintes testes para se qualificar de acordo comAWSPrograma de qualificação do dispositivo.

- Certificado de servidor de nome de assunto incorreto do TLS (p. 1095)(“Nome comum do assunto incorreto (CN)/Nome alternativo do assunto (SAN)”)
- Certificado de servidor não seguro TLS (p. 1094)(“Não assinado pela CA reconhecida”)
- Connect TLS (p. 1092)(“Connect TLS”)
- MQTT Connect (p. 1095)(“Dispositivo envia CONNECT paraAWS IoT Core(Caso feliz)”)
- MQTT Subscribe (p. 1099)(“Pode se inscrever (caso feliz)”)
- MQTT Publish (p. 1098)(“QoS0 (Caso feliz)”)

TLS

Você usa esses testes para determinar se o protocolo de segurança da camada de transporte (TLS) entre seus dispositivos eAWS IoTÉ seguro.

Caminho feliz

"Connect TLS"

Valida se o dispositivo em teste pode completar o handshake TLS paraAWS IoT. Este teste não valida a implementação do MQTT do dispositivo cliente.

Definição de caso de teste de API:

> **Note**
>
> `EXECUTION_TIMEOUT`O tem um valor padrão de 5 minutos. Para melhores resultados, recomendamos um valor de tempo limite de 2 minutos.

```
"tests":[
   {
      "name":"my_tls_connect_test",
      "configuration": {
         // optional:
         "EXECUTION_TIMEOUT":"300",  //in seconds
      },
      "test":{
         "id":"TLS_Connect",
         "version":"0.0.0"
      }
   }
]
```

Example Saídas de caso de teste:

- Pass— O dispositivo em teste concluiu o aperto de mão TLS comAWS IoT.
- Passe com avisos— O dispositivo em teste concluiu o aperto de mão TLS comAWS IoT, mas havia mensagens de aviso TLS do dispositivo ouAWS IoT.
- Fail— O dispositivo em teste não conseguiu concluir o aperto de mão TLS comAWS IoTdevido a erro de aperto de mão.

"Fragmentos de tamanho máximo de recebimento de TLS"

Este caso de teste ajuda a validar se o dispositivo pode receber e processar fragmentos de tamanho máximo de TLS. Seu dispositivo de teste precisa se inscrever em um tópico pré-configurado com QoS 1 para receber uma grande carga útil. Você pode personalizar a carga usando a configuração ${payload}.

Definição de caso de teste de API:

> **Note**
>
> `EXECUTION_TIMEOUT`O tem um valor padrão de 5 minutos. Para melhores resultados, recomendamos um valor de tempo limite de 2 minutos.

```
"tests":[
   {
      "name":"TLS Receive Maximum Size Fragments",
      "configuration": {
         // optional:
         "EXECUTION_TIMEOUT":"300",  //in seconds
         "PAYLOAD_FORMAT":"{"message":"${payload}"}", // A string with a placeholder
 ${payload}, or leave it empty to receive a plain string.
         "TRIGGER_TOPIC": "test_1" // A topic to which a device will subscribe, and to
 which a test case will publish a large payload.
```

```
      },
      "test":{
         "id":"TLS_Receive_Maximum_Size_Fragments",
         "version":"0.0.0"
      }
   }
]
```

Pacotes de criptografia

“Support a dispositivos TLS paraAWS IoTPacotes de criptografia recomendados”

Valida que os conjuntos de cifras na mensagem de saudação do cliente TLS do dispositivo em teste contémAWS IoTPacotes de criptografia recomendados (p. 381). Ele fornece informações adicionais sobre conjuntos de cifras compatíveis com o dispositivo.

Definição de caso de teste de API:

> **Note**
>
> `EXECUTION_TIMEOUT`O tem um valor padrão de 5 minutos. Recomendamos um valor de tempo limite de 2 minutos.

```
"tests":[
   {
      "name":"my_tls_support_aws_iot_cipher_suites_test",
      "configuration": {
         // optional:
         "EXECUTION_TIMEOUT":"300",  // in seconds
      },
      "test":{
         "id":"TLS_Support_AWS_IoT_Cipher_Suites",
         "version":"0.0.0"
      }
   }
]
```

Example Saídas de caso de teste:

- Pass— O dispositivo em conjuntos de cifras de teste contém pelo menos umAWS IoTpacote de cifras recomendado e não contém conjuntos de cifras sem suporte.
- Passe com avisos— Os conjuntos de cifras do dispositivo contêm pelo menos umAWS IoTconjunto de cifras, mas 1) não contém nenhum dos conjuntos de cifras recomendados, ou 2) contém conjuntos de cifras não suportados porAWS IoT. Sugerimos verificar se os conjuntos de cifras não suportados são seguros.
- Fail— O dispositivo em conjuntos de cifras de teste não contém nenhum dosAWS IoTPacotes de criptografia compatíveis.

Certificado de servidor de tamanho maior

“Certificado de servidor de tamanho grande TLS”

Valida se o dispositivo pode concluir o handshake TLS comAWS IoTrecebendo e processando um certificado de servidor de tamanho maior. O tamanho do certificado do servidor (em bytes), usado por esse teste é maior do que o usado atualmente na“Connect TLS”caso de teste e IoT Core em 20%. Durante este caso de teste, o espaço de buffer do dispositivo para TLS será testado. Se o espaço do buffer for grande o suficiente, o handshake TLS será concluído sem erros. Este teste não valida a

implementação do MQTT do dispositivo. O caso de teste termina após o processo de handshake TLS ser concluído.

Definição de caso de teste de API:

> **Note**
>
> `EXECUTION_TIMEOUT`O tem um valor padrão de 5 minutos. Para melhores resultados, recomendamos um valor de tempo limite de 2 minutos. Se este caso de teste falhar, mas“Connect TLS”caso de teste passa, recomendamos que você aumente o limite de espaço de buffer do seu dispositivo para TLS para um número maior de bytes. Aumentar o limite de espaço de buffer garantirá que seu dispositivo seja capaz de processar um certificado de servidor de tamanho maior caso o tamanho aumente.

```
"tests":[
   {
      "name":"my_tls_large_size_server_cert_test",
      "configuration": {
         // optional:
         "EXECUTION_TIMEOUT":"300",  // in seconds
      },
      "test":{
         "id":"TLS_Large_Size_Server_Cert",
         "version":"0.0.0"
      }
   }
]
```

Example Saídas de caso de teste:

- Pass— O dispositivo em teste completou o aperto de mão TLS comAWS IoT.
- Passe com avisos— O dispositivo em teste completou o aperto de mão TLS comAWS IoT, mas há mensagens de aviso TLS do dispositivo ouAWS IoT.
- Fail— O dispositivo em teste não conseguiu concluir o aperto de mão TLS comAWS IoTdevido a um erro durante o processo de handshake.

Certificado de servidor não seguro TLS

“Não assinado pela CA reconhecida”

Valida se o dispositivo em teste fecha a conexão se for apresentado com um certificado de servidor que não tenha uma assinatura válida da CA ATS. Um dispositivo só deve se conectar a um endpoint que apresente um certificado válido.

Definição de caso de teste de API:

> **Note**
>
> `EXECUTION_TIMEOUT`O tem um valor padrão de 5 minutos. Recomendamos um valor de tempo limite de 2 minutos.

```
"tests":[
   {
      "name":"my_tls_unsecure_server_cert_test",
      "configuration": {
         // optional:
         "EXECUTION_TIMEOUT":"300",  //in seconds
      },
      "test":{
         "id":"TLS_Unsecure_Server_Cert",
```

```
            "version":"0.0.0"
         }
      }
]
```

Example Saídas de caso de teste:

- Pass— O dispositivo em teste fechou a conexão.
- Fail— O dispositivo em teste concluiu o aperto de mão TLS comAWS IoT.

"Certificado de servidor de nome de assunto incorreto do TLS/Nome comum (CN) do assunto incorreto (CN)/Nome alternativo do assunto (SAN)"

Valida se o dispositivo em teste fecha a conexão se for apresentado com um certificado de servidor para um nome de domínio diferente do solicitado.

Definição de caso de teste de API:

> **Note**
>
> `EXECUTION_TIMEOUT`O tem um valor padrão de 5 minutos. Recomendamos um valor de tempo limite de 2 minutos.

```
"tests":[
   {
      "name":"my_tls_incorrect_subject_name_cert_test",
      "configuration": {
         // optional:
         "EXECUTION_TIMEOUT":"300",   // in seconds
      },
      "test":{
         "id":"TLS_Incorrect_Subject_Name_Server_Cert",
         "version":"0.0.0"
      }
   }
]
```

Example Saídas de caso de teste:

- Pass— O dispositivo em teste fechou a conexão.
- Fail— O dispositivo em teste concluiu o aperto de mão TLS comAWS IoT.

MQTT

CONECTAR, DESCONECTAR E RECONECTAR

"Dispositivo envia CONNECT paraAWS IoT Core(Estojo feliz)"

Valida se o dispositivo em teste envia uma solicitação CONNECT.

Definição de caso de teste de API:

> **Note**
>
> `EXECUTION_TIMEOUT`O tem um valor padrão de 5 minutos. Recomendamos um valor de tempo limite de 2 minutos.

```
"tests":[
   {
```

```
      "name":"my_mqtt_connect_test",
      "configuration": {
         // optional:
         "EXECUTION_TIMEOUT":"300",   // in seconds
      },
      "test":{
         "id":"MQTT_Connect",
         "version":"0.0.0"
      }
   }
]
```

“O dispositivo pode retornar o PUBACK a um tópico arbitrário para QoS1"

Este caso de teste verificará se o dispositivo (cliente) pode retornar uma mensagem PUBACK se receber uma mensagem de publicação do corretor depois de se inscrever em um tópico com QoS1.

Definição de caso de teste de API:

Note

`EXECUTION_TIMEOUT`O tem um valor padrão de 5 minutos. Recomendamos um valor de tempo limite de 2 minutos.

```
"tests":[
   {
        "name":"my_mqtt_client_puback_qos1",
        "configuration": {
            // optional:
            "TRIGGER_TOPIC": "myTopic",
            "EXECUTION_TIMEOUT":"300", // in seconds
            "PAYLOAD_FOR_PUBLISH_VALIDATION":"custom payload"
        },
        "test": {
            "id": "MQTT_Client_Puback_Qos1",
            "version": "0.0.0"
        }
    }
]
```

“Retentativas de conexão do dispositivo com recuo de jitter - Sem resposta CONNACK”

Valida que o dispositivo em teste usa o recuo de jitter adequado ao reconectar com o corretor por pelo menos cinco vezes. O corretor registra o carimbo de data/hora do dispositivo na solicitação CONNECT do teste, executa a validação de pacotes, faz uma pausa sem enviar um CONNACK para o dispositivo em teste e aguarda que o dispositivo em teste reenvie a solicitação. A sexta tentativa de conexão pode passar e CONNACK pode fluir de volta para o dispositivo em teste.

O processo anterior é executado novamente. No total, este caso de teste requer que o dispositivo se conecte pelo menos 12 vezes no total. Os carimbos de data/hora coletados são usados para validar que o recuo de jitter é usado pelo dispositivo em teste. Se o dispositivo em teste tiver um atraso de recuo estritamente exponencial, este caso de teste passará com avisos.

Recomendamos a implementação doRetorno exponencial e jitterMecanismo no dispositivo em teste para passar neste caso de teste.

Definição de caso de teste de API:

Note

`EXECUTION_TIMEOUT`O tem um valor padrão de 5 minutos. Recomendamos um valor de tempo limite de 4 minutos.

```
"tests":[
   {
      "name":"my_mqtt_jitter_backoff_retries_test",
      "configuration": {
         // optional:
         "EXECUTION_TIMEOUT":"300",    // in seconds
      },
      "test":{
         "id":"MQTT_Connect_Jitter_Backoff_Retries",
         "version":"0.0.0"
      }
   }
]
```

“Retentativas de conexão do dispositivo com recuo exponencial - Sem resposta CONNACK”

Valida que o dispositivo em teste usa o recuo exponencial adequado ao reconectar com o corretor por pelo menos cinco vezes. O corretor registra o carimbo de data/hora do dispositivo na solicitação CONNECT do teste, executa a validação de pacotes, faz uma pausa sem enviar um CONNACK para o dispositivo cliente e aguarda que o dispositivo em teste reenvie a solicitação. Os carimbos de data/hora coletados são usados para validar que um recuo exponencial é usado pelo dispositivo em teste.

Recomendamos a implementação doRetorno exponencial e jitterMecanismo no dispositivo em teste para passar neste caso de teste.

Definição de caso de teste de API:

> **Note**
>
> `EXECUTION_TIMEOUT`O tem um valor padrão de 5 minutos. Recomendamos um valor de tempo limite de 4 minutos.

```
"tests":[
   {
      "name":"my_mqtt_exponential_backoff_retries_test",
      "configuration": {
         // optional:
         "EXECUTION_TIMEOUT":"600",  // in seconds
      },
      "test":{
         "id":"MQTT_Connect_Exponential_Backoff_Retries",
         "version":"0.0.0"
      }
   }
]
```

“Reconexão do dispositivo com pelo menos um segundo atraso - Após a desconexão do servidor”

Valida se um dispositivo em teste aguarda pelo menos um segundo antes de se reconectar ao servidor quando o Device Advisor desconecta imediatamente o dispositivo após uma conexão bem-sucedida. Se o dispositivo tentar se reconectar em menos de um segundo, o teste falhará.

Definição de caso de teste de API:

> **Note**
>
> `EXECUTION_TIMEOUT`O tem um valor padrão de 5 minutos. Recomendamos um valor de tempo limite de 2 minutos.

```
"tests":[
   {
      "name":"my_mqtt_reconnection_after_successful_connection",
```

```
      "configuration":{
        // optional:
        "EXECUTION_TIMEOUT":"300"  //in seconds
      },
      "test":{
         "id":"MQTT_ReconnectionAfterSuccessfulConnection",
         "version":"0.0.0"
      }
   }
]
```

“Reconexão do dispositivo com recuo de jitter - Após a desconexão do servidor”

Valida se um dispositivo em teste usa o jitter e o recuo necessários durante a reconexão após a desconexão do servidor. O Device Advisor desconecta o dispositivo do servidor por pelo menos cinco vezes e observa o comportamento do dispositivo para a reconexão MQTT. O Device Advisor registra o carimbo de data e hora da solicitação CONNECT para o dispositivo em teste, executa a validação de pacotes, faz uma pausa sem enviar um CONNACK para o dispositivo cliente e aguarda que o dispositivo em teste reenvie a solicitação. Os carimbos de data/hora coletados são usados para validar se o dispositivo em teste usa jitter e recuo durante a reconexão. Se o dispositivo em teste tiver um recuo estritamente exponencial ou não implementar um mecanismo de recuo de jitter adequado, esse caso de teste passará com avisos. Se o dispositivo em teste tiver implementado um recuo linear ou um mecanismo de recuo constante, o teste falhará.

Para passar neste caso de teste, recomendamos implementar oRetorno exponencial e jitterMecanismo no dispositivo em teste.

Definição de caso de teste de API:

> **Note**
>
> `EXECUTION_TIMEOUT`O tem um valor padrão de 5 minutos. Recomendamos um valor de tempo limite de 4 minutos.

```
"tests":[
   {
      "name":"my_mqtt_reconnect_backoff_retries_on_server_disconnect",
      "configuration":{
         // optional:
         "EXECUTION_TIMEOUT":"300",  // in seconds
         "RECONNECTION_ATTEMPTS": 5
      },
      "test":{
         "id":"MQTT_Reconnect_Backoff_Retries_On_Server_Disconnect",
         "version":"0.0.0"
      }
   }
]
```

O número de tentativas de reconexão para validar para recuo pode ser alterado especificando o`RECONNECTION_ATTEMPTS`. O número deve estar entre cinco e dez. O valor padrão é cinco.

Publicar

“QoS0 (Caso feliz)”

Valida que o dispositivo em teste publica uma mensagem com QoS0. Você também pode validar o tópico da mensagem especificando esse valor de tópico nas configurações de teste.

Definição de caso de teste de API:

> **Note**
>
> `EXECUTION_TIMEOUT`O tem um valor padrão de 5 minutos. Recomendamos um valor de tempo limite de 2 minutos.

```
"tests":[
   {
      "name":"my_mqtt_publish_test",
      "configuration":{
         // optional:
         "EXECUTION_TIMEOUT":"300",  // in seconds
         "TOPIC_FOR_PUBLISH_VALIDATION": "my_TOPIC_FOR_PUBLISH_VALIDATION",
         "PAYLOAD_FOR_PUBLISH_VALIDATION": "my_PAYLOAD_FOR_PUBLISH_VALIDATION",
      },
      "test":{
         "id":"MQTT_Publish",
         "version":"0.0.0"
      }
   }
]
```

“Repetição de publicação QoS1 - Sem PUBACK”

Valida se o dispositivo em teste republica uma mensagem enviada com QoS1, se o corretor não enviar PUBACK. Você também pode validar o tópico da mensagem especificando esse tópico nas configurações de teste. O dispositivo cliente não deve se desconectar antes de republicar a mensagem. Este teste também valida que a mensagem republicada tem o mesmo identificador de pacote que o original.

Definição de caso de teste de API:

> **Note**
>
> `EXECUTION_TIMEOUT`O tem um valor padrão de 5 minutos. É recomendado por pelo menos 4 minutos.

```
"tests":[
   {
      "name":"my_mqtt_publish_retry_test",
      "configuration":{
         // optional:
         "EXECUTION_TIMEOUT":"300",  // in seconds
         "TOPIC_FOR_PUBLISH_VALIDATION": "my_TOPIC_FOR_PUBLISH_VALIDATION",
         "PAYLOAD_FOR_PUBLISH_VALIDATION": "my_PAYLOAD_FOR_PUBLISH_VALIDATION",
      },
      "test":{
         "id":"MQTT_Publish_Retry_No_Puback",
         "version":"0.0.0"
      }
   }
]
```

Inscrever-se

“Pode se inscrever (caso feliz)”

Valida que o dispositivo em teste assina tópicos do MQTT. Você também pode validar o tópico que o dispositivo em teste se inscreve especificando esse tópico nas configurações de teste.

Definição de caso de teste de API:

> **Note**
>
> `EXECUTION_TIMEOUT`O tem um valor padrão de 5 minutos. Recomendamos um valor de tempo limite de 2 minutos.

```
"tests":[
   {
      "name":"my_mqtt_subscribe_test",
      "configuration":{
         // optional:
         "EXECUTION_TIMEOUT":"300",  // in seconds
         "TOPIC_LIST_FOR_SUBSCRIPTION_VALIDATION_ID":
["my_TOPIC_FOR_PUBLISH_VALIDATION_a","my_TOPIC_FOR_PUBLISH_VALIDATION_b"]
      },
      "test":{
         "id":"MQTT_Subscribe",
         "version":"0.0.0"
      }
   }
]
```

“Subscrever novamente - Sem SUBACK”

Valida que o dispositivo em teste tenta repetir uma assinatura com falha dos tópicos do MQTT. O servidor então aguarda e não envia um SUBACK. Se o dispositivo cliente não tentar a assinatura novamente, o teste falhará. O dispositivo cliente deve repetir a assinatura com falha com o mesmo ID de pacote. Você também pode validar o tópico que o dispositivo em teste se inscreve especificando esse tópico nas configurações de teste.

Definição de caso de teste de API:

> **Note**
>
> `EXECUTION_TIMEOUT`O tem um valor padrão de 5 minutos. Recomendamos um valor de tempo limite de 4 minutos.

```
"tests":[
   {
      "name":"my_mqtt_subscribe_retry_test",
      "configuration":{
         "EXECUTION_TIMEOUT":"300",  // in seconds
         // optional:
         "TOPIC_LIST_FOR_SUBSCRIPTION_VALIDATION_ID":
["myTOPIC_FOR_PUBLISH_VALIDATION_a","my_TOPIC_FOR_PUBLISH_VALIDATION_b"]
      },
      "test":{
         "id":"MQTT_Subscribe_Retry_No_Suback",
         "version":"0.0.0"
      }
   }
]
```

Keep-alive

“Mqtt Sem Ack PingResp”

Este caso de teste valida se o dispositivo em teste se desconecta quando não receber uma resposta de ping. Como parte desse caso de teste, o Device Advisor bloqueia respostas enviadas deAWS IoT Corepara solicitações de publicação, assinatura e ping. Ele também valida se o dispositivo em teste desconecta a conexão MQTT.

Definição de caso de teste de API:

> **Note**
>
> `EXECUTION_TIMEOUT`O tem um valor padrão de 5 minutos. Recomendamos um tempo limite maior que 1,5 vezes `okeepAliveTime`value.

```
"tests":[
    {
       "name":"Mqtt No Ack PingResp",
       "configuration":
          //optional:
          "EXECUTION_TIMEOUT":"306",   // in seconds
       },
       "test":{
          "id":"MQTT_No_Ack_PingResp",
          "version":"0.0.0"
       }
    }
]
```

Sessão persistente

“Sessão persistente (caso feliz)”

Este caso de teste valida o comportamento do dispositivo quando desconectado de uma sessão persistente. O caso de teste verifica se o dispositivo pode se reconectar, retomar as assinaturas de seus tópicos de gatilho sem se inscrever explicitamente, receber as mensagens armazenadas nos tópicos e funcionar conforme o esperado durante uma sessão persistente. Quando esse caso de teste passa, ele indica que o dispositivo cliente é capaz de manter uma sessão persistente com oAWS IoT Corecorretor de uma maneira esperada. Para obter mais informações sobre asAWS IoTSessões persistentes, consulteUsando sessões persistentes do MQTT.

Neste caso de teste, espera-se que o dispositivo cliente se CONECTE com oAWS IoT Corecom um sinalizador de sessão limpa definido como false e, em seguida, assine um tópico de gatilho. Após uma assinatura bem-sucedida, o dispositivo será desconectado porAWS IoT CoreDevice Advisor. Enquanto o dispositivo estiver em um estado desconectado, uma carga útil de mensagem QoS 1 será armazenada nesse tópico. O Device Advisor permitirá que o dispositivo cliente se reconecte com o endpoint de teste. Neste ponto, como há uma sessão persistente, espera-se que o dispositivo cliente retome suas assinaturas de tópicos sem enviar nenhum pacote SUBSCRIBE adicional e receber a mensagem QoS 1 do corretor. Depois de reconectar, se o dispositivo cliente se inscrever novamente no tópico de gatilho enviando um pacote SUBSCRIBE adicional e/ou se o cliente não receber a mensagem armazenada do tópico do gatilho, o caso de teste falhará.

Definição de caso de teste de API:

> **Note**
>
> `EXECUTION_TIMEOUT`O tem um valor padrão de 5 minutos. Recomendamos um valor de tempo limite de pelo menos 4 minutos. Na primeira conexão, o dispositivo cliente precisa se inscrever explicitamente em um`TRIGGER_TOPIC`que não foi inscrito antes. Para passar no caso de teste, o dispositivo cliente deve se inscrever com sucesso no`TRIGGER_TOPIC`com um QoS 1. Após a reconexão, espera-se que o dispositivo cliente entenda que há uma sessão persistente ativa; portanto, ele deve aceitar a mensagem armazenada enviada pelo tópico do gatilho e retornar PUBACK para essa mensagem específica.

```
"tests":[
   {
```

```
      "name":"my_mqtt_persistent_session_happy_case",
      "configuration":{
         //required:
         "TRIGGER_TOPIC": "myTrigger/topic",
         // optional:
         // if Payload not provided, a string will be stored in the trigger topic to be
 sent back to the client device
         "PAYLOAD": "The message which should be received from AWS IoT Broker after re-
connecting to a persistent session from the specified trigger topic.",
         "EXECUTION_TIMEOUT":"300" // in seconds
      },
      "test":{
         "id":"MQTT_Persistent_Session_Happy_Case",
         "version":"0.0.0"
      }
   }
]
```

“Sessão persistente - Expiração da sessão”

Esse caso de teste ajuda a validar o comportamento do dispositivo quando um dispositivo desconectado se reconecta a uma sessão persistente expirada. Depois que a sessão expirar, esperamos que o dispositivo reenvie para os tópicos anteriormente inscritos enviando explicitamente um novo pacote SUBSCRIBE.

Durante a primeira conexão, esperamos que o dispositivo de teste se CONECTE com oAWSCorretor de IoT, como seu`CleanSession`flag é definido como false para iniciar uma sessão persistente. Em seguida, o dispositivo deve assinar um tópico de gatilho. Em seguida, o dispositivo é desconectado porAWS IoT CoreDevice Advisor, após uma assinatura bem-sucedida e início de uma sessão persistente. Após a desconexão,AWS IoT CoreO Device Advisor permite que o dispositivo de teste se reconecte novamente com o endpoint de teste. Neste ponto, quando o dispositivo de teste envia outro pacote CONNECT,AWS IoT CoreO Device Advisor envia de volta um pacote CONNACK que indica que a sessão persistente expirou. O dispositivo de teste precisa interpretar esse pacote corretamente, e espera-se que se inscreva novamente no mesmo tópico de gatilho novamente à medida que a sessão persistente for encerrada. Se o dispositivo de teste não assinar novamente o gatilho de tópico novamente, o caso de teste falhará. Para que o teste seja aprovado, o dispositivo precisa entender que a sessão persistente acabou e enviar de volta um novo pacote SUBSCRIBE para o mesmo tópico de gatilho na segunda conexão.

Se este caso de teste for aprovado para um dispositivo de teste, ele indica que o dispositivo é capaz de lidar com a reconexão no vencimento da sessão persistente de uma maneira esperada.

Definição de caso de teste de API:

> **Note**
>
> `EXECUTION_TIMEOUT`O tem um valor padrão de 5 minutos. Recomendamos um valor de tempo limite de pelo menos 4 minutos. O dispositivo de teste precisa se inscrever explicitamente em um`TRIGGER_TOPIC`, para o qual não foi inscrito antes. Para passar no caso de teste, o dispositivo de teste deve enviar um pacote CONNECT com`CleanSession`sinalizador definido como false e assine com êxito um tópico de gatilho com um QoS 1. Depois de uma conexão bem-sucedida,AWS IoT CoreO Device Advisor desconecta o dispositivo. Após a desconexão,AWS IoT CoreO Device Advisor permite que o dispositivo se reconecte novamente, e espera-se que o dispositivo se inscreva novamente no mesmo`TRIGGER_TOPIC`Desde aAWS IoT CoreO Device Advisor teria encerrado a sessão persistente.

```
"tests":[
   {
      "name":"my_expired_persistent_session_test",
```

```
      "configuration":{
         //required:
         "TRIGGER_TOPIC": "myTrigger/topic",
         // optional:
         "EXECUTION_TIMEOUT":"300" // in seconds
      },
      "test":{
         "id":"MQTT_Expired_Persistent_Session",
         "version":"0.0.0"
      }
   }
]
```

Shadow

Use esses testes para verificar seus dispositivos em uso de testeAWS IoTServiço Device Shadow corretamente. Consulte Serviço Device Shadow do AWS IoT (p. 633) para obter mais informações. Se esses casos de teste estiverem configurados em seu conjunto de testes, é necessário fornecer uma coisa ao iniciar a execução da suíte.

Publicar

“O dispositivo publica o estado após a conexão (caso feliz)”

Valida se um dispositivo pode publicar seu estado depois de se conectar aoAWS IoT Core

Definição de caso de teste de API:

> **Note**
>
> `EXECUTION_TIMEOUT`O tem um valor padrão de 5 minutos. Recomendamos um valor de tempo limite de 2 minutos.

```
"tests":[
   {
      "name":"my_shadow_publish_reported_state",
      "configuration": {
         // optional:
         "EXECUTION_TIMEOUT":"300", // in seconds
         "SHADOW_NAME": "SHADOW_NAME",
         "REPORTED_STATE": {
            "STATE_ATTRIBUTE": "STATE_VALUE"
         }
      },
      "test":{
         "id":"Shadow_Publish_Reported_State",
         "version":"0.0.0"
      }
   }
]
```

O`REPORTED_STATE`pode ser fornecido para validação adicional no estado de sombra exato do dispositivo, depois que ele se conecta. Por padrão, esse caso de teste valida o estado de publicação do dispositivo.

Se*`SHADOW_NAME`*não é fornecido, o caso de teste procura mensagens publicadas nos prefixos de tópicos do tipo de sombra sem nome (clássico) por padrão. Forneça um nome de sombra se o dispositivo usar o tipo de sombra nomeado. ConsulteUsar sombras em dispositivospara obter mais informações.

Atualização

“Atualizações do dispositivo relatadas estado para o estado desejado (caso feliz)”

Valida se o dispositivo ler todas as mensagens de atualização recebidas e sincronizar o estado do dispositivo para que corresponda às propriedades de estado desejadas. Seu dispositivo deve publicar o último estado relatado após a sincronização. Se o dispositivo já tiver uma sombra existente antes de executar o teste, verifique se o estado desejado configurado para o caso de teste e o estado relatado existente ainda não correspondem. Você pode identificar as mensagens de atualização do Shadow enviadas pelo Device Advisor observando oClientTokencampo no documento Shadow como ele será`DeviceAdvisorShadowTestCaseSetup`.

Definição de caso de teste de API:

> **Note**
>
> `EXECUTION_TIMEOUT`O tem um valor padrão de 5 minutos. Recomendamos um valor de tempo limite de 2 minutos.

```
"tests":[
   {
      "name":"my_shadow_update_reported_state",
      "configuration": {
         "DESIRED_STATE": {
            "STATE_ATTRIBUTE": "STATE_VALUE"
         },
         // optional:
         "EXECUTION_TIMEOUT":"300", // in seconds
         "SHADOW_NAME": "SHADOW_NAME"
      },
      "test":{
         "id":"Shadow_Update_Reported_State",
         "version":"0.0.0"
      }
   }
]
```

O`DESIRED_STATE`Deve ter pelo menos um atributo e valor associado.

Se`SHADOW_NAME`não é fornecido, então o caso de teste procura mensagens publicadas nos prefixos de tópicos do tipo de sombra sem nome (clássico) por padrão. Forneça um nome de sombra se o dispositivo usar o tipo de sombra nomeado. ConsulteUsar sombras em dispositivospara obter mais informações.

Execução de Job

“O dispositivo pode concluir uma execução de trabalho”

Este caso de teste ajuda você a validar se seu dispositivo pode receber atualizações usandoAWS IoTTrabalhos e publique o status das atualizações bem-sucedidas. Para obter mais informações sobre asAWS IoTEmpregos, consulteTrabalhos do.

Definição de caso de teste de API:

> **Note**
>
> `EXECUTION_TIMEOUT`O tem um valor padrão de 5 minutos. Recomendamos um valor de tempo limite de 2 minutos.

```
"tests": [
   {
      "name":"my_job_execution",
      "configuration": {
         // optional:
         // Test case will create a job task by using either JOB_DOCUMENT or
 JOB_DOCUMENT_SOURCE.
         // If you manage the job task on your own, leave it empty and provide the
 JOB_JOBID (self-managed job task).
         // JOB_DOCUMENT is a JSON formatted string
         "JOB_DOCUMENT": "{
            \"operation\":\"reboot\",
            \"files\" : {
               \"fileName\" : \"install.py\",
               \"url\" : \"${aws:iot:s3-presigned-url:https://s3.amazonaws.com/bucket-
name/key}\"
            }
         }",
         // JOB_DOCUMENT_SOURCE is an S3 link to the job document. It will be used only
 if JOB_DOCUMENT is not provided.
         "JOB_DOCUMENT_SOURCE": "https://s3.amazonaws.com/bucket-name/key",
         // JOB_JOBID is mandatory, only if neither document nor document source is
 provided. (Test case needs to know the self-managed job task id).
         "JOB_JOBID": "String",
         // JOB_PRESIGN_ROLE_ARN is used for the presign Url, which will replace the
 placeholder in the JOB_DOCUMENT field
         "JOB_PRESIGN_ROLE_ARN": "String",
         // Presigned Url expiration time. It must be between 60 and 3600 seconds, with
 the default value being 3600.
         "JOB_PRESIGN_EXPIRES_IN_SEC": "Long"
         "EXECUTION_TIMEOUT": "300", // in seconds
      },
      "test": {
         "id": "Job_Execution",
         "version": "0.0.0"
      }
   }
]
```

Para obter mais informações sobre como criar e usar documentos de trabalho, consultedocumento de trabalho.

Políticas e permissões

Você pode usar os testes a seguir para determinar se as políticas anexadas aos certificados de seus dispositivos seguem as melhores práticas padrão.

“As políticas anexadas ao certificado de dispositivo não contêm curingas”

Valida se as políticas de permissão associadas a um dispositivo seguem as melhores práticas e não concedem ao dispositivo mais permissões do que o necessário.

Definição de caso de teste de API:

> **Note**
>
> `EXECUTION_TIMEOUT`O tem um valor padrão de 1 minuto. Recomendamos definir um tempo limite de pelo menos 30 segundos.

```
"tests":[
   {
```

```
        "name":"my_security_device_policies",
        "configuration": {
            // optional:
            "EXECUTION_TIMEOUT":"60"    // in seconds
        },
        "test": {
            "id": "Security_Device_Policies",
            "version": "0.0.0"
        }
    }
]
```

Mensagens de eventos

Esta seção contém informações sobre mensagens publicadas pela AWS IoT quando coisas ou trabalhos são atualizados ou alterados. Para obter informações sobre oAWS IoT EventsServiço que permite criar detectores para monitorar seus dispositivos para encontrar falhas ou alterações na operação e para acionar ações específicas quando elas ocorrerem, consulteAWS IoT Events.

Como as mensagens de eventos são geradas

A AWS IoT publica mensagens de eventos quando ocorrerem determinados eventos. Por exemplo, os eventos são gerados pelo registro quando as coisas são adicionadas, atualizadas ou excluídas. Cada evento faz com que uma única mensagem de evento seja enviada. As mensagens de eventos são publicadas por meio do MQTT com uma carga JSON. O conteúdo da carga depende do tipo do evento.

> **Note**
>
> Há garantia de que as mensagens de eventos sejam publicadas uma vez. É possível que elas sejam publicadas mais de uma vez. A ordenação das mensagens de eventos não é garantida.

Política para receber mensagens de eventos

Para receber mensagens de eventos, seu dispositivo deve usar uma política adequada que permita que ele se conecte ao gateway de dispositivos da AWS IoT e se inscreva em tópicos de eventos MQTT. Você também deve assinar os filtros apropriados dos tópicos.

Veja a seguir um exemplo da política necessária para o recebimento de eventos de ciclo de vida:

```
{
    "Version":"2012-10-17",
    "Statement":[{
        "Effect":"Allow",
        "Action":[
            "iot:Subscribe",
            "iot:Receive"
        ],
        "Resource":[
            "arn:aws:iot:region:account:/$aws/events/*"
        ]
    }]
}
```

Habilite eventos para oAWS IoT

Antes que os assinantes dos tópicos reservados possam receber mensagens, você deve habilitar mensagens de evento a partir doAWS Management Consoleou usando a API ou CLI. Para obter informações sobre as mensagens de eventos que as diferentes opções gerenciam, consulte aTabela deAWS IoTconfigurações de eventos (p. 1108).

- Para habilitar mensagens de evento, vá para oConfigurações Guia doAWS IoTconsole e, em seguida, noMensagens baseadas em eventosseção, escolhaGerenciar eventos. Você pode especificar os eventos que você deseja gerenciar.

- Para controlar quais tipos de eventos são publicados usando a API ou a CLI, chame oUpdateEventConfigurationsAPI ou use oupdate-event-configurationsComando da CLI. Por exemplo:

```
aws iot update-event-configurations --event-configurations "{\"THING\":{\"Enabled\":
 true}}"
```

> **Note**
>
> Todas as aspas (") são recuadas com uma barra invertida (\).

Você pode obter a configuração atual do evento chamando a API DescribeEventConfigurations ou com o comando describe-event-configurations da CLI. Por exemplo: .

```
aws iot describe-event-configurations
```

Tabela deAWS IoTconfigurações de eventos

Categoria de eventos (AWS IoTConsole: Configurações: Mensagens baseadas em eventos)	**`eventConfigurations`**Valor da chave (AWS CLI/API)	Tópico de mensagem de eventos
(Só pode ser configurado usando oAWS CLI/API)	`CA_CERTIFICATE`	`$aws/events/certificates/registered/`*`caCertificateId`*
(Só pode ser configurado usando oAWS CLI/API)	`CERTIFICATE`	`$aws/events/presence/connected/`*`clientId`*
(Só pode ser configurado usando oAWS CLI/API)	`CERTIFICATE`	`$aws/events/presence/disconnected/`*`clientId`*
(Só pode ser configurado usando oAWS CLI/API)	`CERTIFICATE`	`$aws/events/subscriptions/subscribed/`*`clientId`*
(Só pode ser configurado usando oAWS CLI/API)	`CERTIFICATE`	`$aws/events/subscriptions/unsubscribed/`*`clientId`*
Job concluído, cancelado	`JOB`	`$aws/events/job/`*`jobID`*`/canceled`
Job concluído, cancelado	`JOB`	`$aws/events/job/`*`jobID`*`/cancellation_in_progress`
Job concluído, cancelado	`JOB`	`$aws/events/job/`*`jobID`*`/completed`
Job concluído, cancelado	`JOB`	`$aws/events/job/`*`jobID`*`/deleted`
Job concluído, cancelado	`JOB`	`$aws/events/job/`*`jobID`*`/deletion_in_progress`

Categoria de eventos (AWS IoTConsole: Configurações: Mensagens baseadas em eventos)	**eventConfigurations**Valor da chave (AWS CLI/API)	Tópico de mensagem de eventos
Execução do Job: sucesso, falha, rejeitada, cancelada, removida	`JOB_EXECUTION`	`$aws/events/jobExecution/`*`jobID`*`/canceled`
Execução do Job: sucesso, falha, rejeitada, cancelada, removida	`JOB_EXECUTION`	`$aws/events/jobExecution/`*`jobID`*`/deleted`
Execução do Job: sucesso, falha, rejeitada, cancelada, removida	`JOB_EXECUTION`	`$aws/events/jobExecution/`*`jobID`*`/failed`
Execução do Job: sucesso, falha, rejeitada, cancelada, removida	`JOB_EXECUTION`	`$aws/events/jobExecution/`*`jobID`*`/rejected`
Execução do Job: sucesso, falha, rejeitada, cancelada, removida	`JOB_EXECUTION`	`$aws/events/jobExecution/`*`jobID`*`/removed`
Execução do Job: sucesso, falha, rejeitada, cancelada, removida	`JOB_EXECUTION`	`$aws/events/jobExecution/`*`jobID`*`/succeeded`
Execução do Job: sucesso, falha, rejeitada, cancelada, removida	`JOB_EXECUTION`	`$aws/events/jobExecution/`*`jobID`*`/timed_out`
Coisa: criada, atualizada, excluída	`THING`	`$aws/events/thing/`*`thingName`*`/created`
Coisa: criada, atualizada, excluída	`THING`	`$aws/events/thing/`*`thingName`*`/updated`
Coisa: criada, atualizada, excluída	`THING`	`$aws/events/thing/`*`thingName`*`/deleted`
Grupo de coisas: adicionado, removido	`THING_GROUP`	`$aws/events/thingGroup/`*`thingGroupName`*`/created`
Grupo de coisas: adicionado, removido	`THING_GROUP`	`$aws/events/thingGroup/`*`thingGroupName`*`/updated`
Grupo de coisas: adicionado, removido	`THING_GROUP`	`$aws/events/thingGroup/`*`thingGroupName`*`/deleted`

Categoria de eventos (AWS IoTConsole: Configurações: Mensagens baseadas em eventos)	**eventConfigurations**Valor da chave (AWS CLI/API)	Tópico de mensagem de eventos
Hierarquia de grupo de coisas: adicionada, removida	`THING_GROUP_HIERARCHY`	`$aws/events/thingGroupHierarchy/thingGroup/`*`parentThingGroupName`*`/childThingGroup/`*`childThingGroupName`*`/added`
Hierarquia de grupo de coisas: adicionada, removida	`THING_GROUP_HIERARCHY`	`$aws/events/thingGroupHierarchy/thingGroup/`*`parentThingGroupName`*`/childThingGroup/`*`childThingGroupName`*`/removed`
Associação ao grupo de coisas: adicionada, removida	`THING_GROUP_MEMBERSHIP`	`$aws/events/thingGroupMembership/thingGroup/`*`thingGroupName`*`/thing/`*`thingName`*`/added`
Associação ao grupo de coisas: adicionada, removida	`THING_GROUP_MEMBERSHIP`	`$aws/events/thingGroupMembership/thingGroup/`*`thingGroupName`*`/thing/`*`thingName`*`/removed`
Tipo de coisa: criado, atualizado, excluído	`THING_TYPE`	`$aws/events/thingType/`*`thingTypeName`*`/created`
Tipo de coisa: criado, atualizado, excluído	`THING_TYPE`	`$aws/events/thingType/`*`thingTypeName`*`/updated`
Tipo de coisa: criado, atualizado, excluído	`THING_TYPE`	`$aws/events/thingType/`*`thingTypeName`*`/deleted`
Associação de tipo de coisa: adicionada, removida	`THING_TYPE_ASSOCIATION`	`$aws/events/thingTypeAssociation/thing/`*`thingName/thingTypeName`

Eventos de registro

O registro pode publicar mensagens de eventos quando coisas, tipos de coisas e grupos de coisas são criados, atualizados ou excluídos. Esses eventos, no entanto, não estão disponíveis por padrão. Para obter informações sobre como ativar esses eventos, consulteHabilite eventos para oAWS IoT (p. 1107).

O registro pode fornecer os seguintes tipos de evento:

- Eventos de coisas (p. 1111)
- Eventos do tipo coisa (p. 1112)
- Eventos de grupo de coisas (p. 1114)

Eventos de coisas

Coisa criada/atualizada/excluída

O registro publica as seguintes mensagens de eventos quando as coisas são criadas, atualizadas ou excluídas:

- `$aws/events/thing/thingName/created`
- `$aws/events/thing/thingName/updated`
- `$aws/events/thing/thingName/deleted`

As mensagens contêm os seguintes exemplos de carga:

```
{
    "eventType" : "THING_EVENT",
    "eventId" : "f5ae9b94-8b8e-4d8e-8c8f-b3266dd89853",
    "timestamp" : 1234567890123,
    "operation" : "CREATED|UPDATED|DELETED",
    "accountId" : "123456789012",
    "thingId" : "b604f69c-aa9a-4d4a-829e-c480e958a0b5",
    "thingName" : "MyThing",
    "versionNumber" : 1,
    "thingTypeName" : null,
    "attributes": {
                "attribute3": "value3",
                "attribute1": "value1",
                "attribute2": "value2"
    }
}
```

As cargas contêm os seguintes atributos:

eventType

Defina como "THING_EVENT".

eventId

Um ID de evento exclusivo (sequência).

timestamp

A data e hora do UNIX de quando o evento ocorreu.

operação

A operação que acionou o evento. Os valores válidos são:

- CREATED
- UPDATED
- DELETED

accountId

SuasConta da AWSID.

thingId

O ID da coisa que está sendo criada, atualizada ou excluída.

thingName

O nome da coisa que está sendo criada, atualizada ou excluída.

versionNumber

A versão da coisa que está sendo criada, atualizada ou excluída. Esse valor é definido como 1 quando uma coisa é criada. Ele é incrementado em 1 toda vez que a coisa é atualizada.

thingTypeName

O tipo da coisa associado à coisa, se existir. Caso contrário, `null`.

attributes

Uma coleção de pares nome-valor associados à coisa.

Eventos do tipo coisa

Eventos relacionados ao tipo de coisa:

- Tipo de coisa criado/suspenso/com suspensão cancelada/excluído (p. 1112)
- Tipo de coisa associado ou desassociado de uma coisa (p. 1113)

Tipo de coisa criado/suspenso/com suspensão cancelada/excluído

O registro publica as seguintes mensagens de eventos quando os tipos de coisas são criados, suspensos, têm a suspensão cancelada ou são excluídos:

- `$aws/events/thingType/`*`thingTypeName`*`/created`
- `$aws/events/thingType/`*`thingTypeName`*`/updated`
- `$aws/events/thingType/`*`thingTypeName`*`/deleted`

A mensagem contém as seguintes cargas de exemplo:

```
{
    "eventType" : "THING_TYPE_EVENT",
    "eventId" : "8827376c-4b05-49a3-9b3b-733729df7ed5",
    "timestamp" : 1234567890123,
    "operation" : "CREATED|UPDATED|DELETED",
    "accountId" : "123456789012",
    "thingTypeId" : "c530ae83-32aa-4592-94d3-da29879d1aac",
    "thingTypeName" : "MyThingType",
    "isDeprecated" : false|true,
    "deprecationDate" : null,
    "searchableAttributes" : [ "attribute1", "attribute2", "attribute3" ],
    "description" : "My thing type"
}
```

As cargas contêm os seguintes atributos:

eventType

Defina como "THING_TYPE_EVENT".

eventId

Um ID de evento exclusivo (sequência).

timestamp

A data e hora do UNIX de quando o evento ocorreu.

operação

A operação que acionou o evento. Os valores válidos são:

- CREATED
- UPDATED
- DELETED

accountId

SuasConta da AWSID.

thingTypeId

O ID do tipo de coisa que está sendo criado, atualizado ou excluído.

thingTypeName

O nome do tipo de coisa que está sendo criado, atualizado ou excluído.

isDeprecated

`true` se o tipo da coisa estiver obsoleto. Caso contrário, `false`.

deprecationDate

A data e hora do UNIX em que o tipo de coisa foi obsoletado.

searchableAttributes

Uma coleção de pares nome-valor associados ao tipo de coisa que pode ser usada para pesquisa.

descrição

Uma descrição do tipo de coisa.

Tipo de coisa associado ou desassociado de uma coisa

O registro publica as seguintes mensagens de eventos quando um tipo de coisa é associado ou desassociado de uma coisa.

- `$aws/events/thingTypeAssociation/thing/`*`thingName`*`/`*`typeName`*

As mensagens contêm os seguintes exemplos de carga:

```
{
    "eventId" : "87f8e095-531c-47b3-aab5-5171364d138d",
    "eventType" : "THING_TYPE_ASSOCIATION_EVENT",
    "operation" : "CREATED|DELETED",
    "thingId" : "b604f69c-aa9a-4d4a-829e-c480e958a0b5",
    "thingName": "myThing",
    "thingTypeName" : "MyThingType",
    "timestamp" : 1234567890123,
}
```

As cargas contêm os seguintes atributos:

eventId

Um ID de evento exclusivo (sequência).

eventType

Defina como "THING_TYPE_ASSOCIATION_EVENT".

operação

A operação que acionou o evento. Os valores válidos são:

- CREATED
- DELETED

thingId

O ID da coisa cuja associação do tipo foi alterada.

thingName

O nome da coisa cuja associação do tipo foi alterada.

thingTypeName

O tipo de coisa associado ou não mais associado à coisa.

timestamp

A data e hora do UNIX de quando o evento ocorreu.

Eventos de grupo de coisas

Eventos relacionados ao grupo de coisas:

- Grupo de coisas criado/atualizado/excluído (p. 1114)
- Coisa adicionada ou removida de um grupo de coisas (p. 1116)
- Grupo de coisas adicionado ou removido de um grupo de coisas (p. 1117)

Grupo de coisas criado/atualizado/excluído

O registro publica as seguintes mensagens de eventos quando um grupo de coisas é criado, atualizado ou excluído.

- `$aws/events/thingGroup/`*`groupName`*`/created`
- `$aws/events/thingGroup/`*`groupName`*`/updated`
- `$aws/events/thingGroup/`*`groupName`*`/deleted`

Veja a seguir um exemplo de um`updated`carga útil. Cargas úteis para`created`e`deleted`as mensagens são semelhantes.

```
{
  "eventType": "THING_GROUP_EVENT",
  "eventId": "8b9ea8626aeaa1e42100f3f32b975899",
  "timestamp": 1603995417409,
  "operation": "UPDATED",
  "accountId": "571EXAMPLE833",
  "thingGroupId": "8757eec8-bb37-4cca-a6fa-403b003d139f",
  "thingGroupName": "Tg_level5",
  "versionNumber": 3,
```

```
  "parentGroupName": "Tg_level4",
  "parentGroupId": "5fce366a-7875-4c0e-870b-79d8d1dce119",
  "description": "New description for Tg_level5",
  "rootToParentThingGroups": [
    {
      "groupArn": "arn:aws:iot:us-west-2:571EXAMPLE833:thinggroup/TgTopLevel",
      "groupId": "36aa0482-f80d-4e13-9bff-1c0a75c055f6"
    },
    {
      "groupArn": "arn:aws:iot:us-west-2:571EXAMPLE833:thinggroup/Tg_level1",
      "groupId": "bc1643e1-5a85-4eac-b45a-92509cbe2a77"
    },
    {
      "groupArn": "arn:aws:iot:us-west-2:571EXAMPLE833:thinggroup/Tg_level2",
      "groupId": "0476f3d2-9beb-48bb-ae2c-ea8bd6458158"
    },
    {
      "groupArn": "arn:aws:iot:us-west-2:571EXAMPLE833:thinggroup/Tg_level3",
      "groupId": "1d9d4ffe-a6b0-48d6-9de6-2e54d1eae78f"
    },
    {
      "groupArn": "arn:aws:iot:us-west-2:571EXAMPLE833:thinggroup/Tg_level4",
      "groupId": "5fce366a-7875-4c0e-870b-79d8d1dce119"
    }
  ],
  "attributes": {
    "attribute1": "value1",
    "attribute3": "value3",
    "attribute2": "value2"
  },
  "dynamicGroupMappingId": null
}
```

As cargas contêm os seguintes atributos:

eventType

Defina como "THING_GROUP_EVENT".

eventId

Um ID de evento exclusivo (sequência).

timestamp

A data e hora do UNIX de quando o evento ocorreu.

operação

A operação que acionou o evento. Os valores válidos são:

- CREATED
- UPDATED
- DELETED

accountId

SuasConta da AWSID.

thingGroupId

O ID do grupo de coisas que está sendo criado, atualizado ou excluído.

thingGroupName

O nome do grupo de coisas que está sendo criado, atualizado ou excluído.

versionNumber

A versão do grupo de coisas. Esse valor é definido como 1 quando um grupo de coisas é criado. Ele é incrementado em 1 toda vez que o grupo de coisas é atualizado.

parentGroupName

O nome do grupo de coisas pai, se houver.

parentGroupId

O ID do grupo de coisas pai, se houver.

descrição

Uma descrição do grupo de coisas.

rootToParentThingGroups

Um conjunto de informações sobre o grupo de coisas pai. Há um elemento para cada grupo de coisas pai, começando pelo grupo de coisas raiz e continuando para o pai do grupo de coisas. Cada entrada contém `ogroupArnegroupId`.

attributes

Uma coleção de pares nome-valor associados ao grupo de coisas.

Coisa adicionada ou removida de um grupo de coisas

O registro publica as seguintes mensagens de eventos quando uma coisa é adicionada ou removida de um grupo de coisas.

- `$aws/events/thingGroupMembership/thingGroup/`*`thingGroupName`*`/thing/`*`thingName`*`/added`
- `$aws/events/thingGroupMembership/thingGroup/`*`thingGroupName`*`/thing/`*`thingName`*`/removed`

As mensagens contêm os seguintes exemplos de carga:

```
{
    "eventType" : "THING_GROUP_MEMBERSHIP_EVENT",
    "eventId" : "d684bd5f-6f6e-48e1-950c-766ac7f02fd1",
    "timestamp" : 1234567890123,
    "operation" : "ADDED|REMOVED",
    "accountId" : "123456789012",
    "groupArn" : "arn:aws:iot:ap-northeast-2:123456789012:thinggroup/MyChildThingGroup",
    "groupId" : "06838589-373f-4312-b1f2-53f2192291c4",
    "thingArn" : "arn:aws:iot:ap-northeast-2:123456789012:thing/MyThing",
    "thingId" : "b604f69c-aa9a-4d4a-829e-c480e958a0b5",
    "membershipId" : "8505ebf8-4d32-4286-80e9-c23a4a16bbd8"
}
```

As cargas contêm os seguintes atributos:

eventType

Defina como "THING_GROUP_MEMBERSHIP_EVENT".

eventId

O ID do evento.

timestamp

A data e hora do UNIX de quando o evento ocorreu.

operação

`ADDED` quando uma coisa é adicionada a um grupo de coisas. `REMOVED` quando uma coisa é removida de um grupo de coisas.

accountId

SuasConta da AWSID.

groupArn

O ARN do grupo de coisas.

groupId

O ID do group.

thingArn

O ARN da coisa que foi adicionada ou removida do grupo de coisas.

thingId

O ID da coisa que foi adicionada ou removida do grupo de coisas.

membershipId

Um ID que representa a relação entre a coisa e o grupo de coisas. Esse valor é gerado quando você adiciona uma coisa a um grupo de coisas.

Grupo de coisas adicionado ou removido de um grupo de coisas

O registro publica as seguintes mensagens de eventos quando um grupo de coisas é adicionado ou removido de um outro grupo de coisas.

- `$aws/events/thingGroupHierarchy/thingGroup/`*`parentThingGroupName`*`/childThingGroup/`*`childThingGroupName`*`/added`
- `$aws/events/thingGroupHierarchy/thingGroup/`*`parentThingGroupName`*`/childThingGroup/`*`childThingGroupName`*`/removed`

A mensagem contém as seguintes cargas de exemplo:

```
{
    "eventType" : "THING_GROUP_HIERARCHY_EVENT",
    "eventId" : "264192c7-b573-46ef-ab7b-489fcd47da41",
    "timestamp" : 1234567890123,
    "operation" : "ADDED|REMOVED",
    "accountId" : "123456789012",
    "thingGroupId" : "8f82a106-6b1d-4331-8984-a84db5f6f8cb",
    "thingGroupName" : "MyRootThingGroup",
    "childGroupId" : "06838589-373f-4312-b1f2-53f2192291c4",
    "childGroupName" : "MyChildThingGroup"
}
```

As cargas contêm os seguintes atributos:

eventType

Defina como "THING_GROUP_HIERARCHY_EVENT".

eventId

O ID do evento.

timestamp

A data e hora do UNIX de quando o evento ocorreu.

operação

`ADDED` quando uma coisa é adicionada a um grupo de coisas. `REMOVED` quando uma coisa é removida de um grupo de coisas.

accountId

SuasConta da AWSID.

thingGroupId

O ID do grupo de coisas pai.

thingGroupName

O nome do grupo de coisas pai.

childGroupId

O ID do grupo de coisas filho.

childGroupName

O nome do grupo de coisas filho.

Eventos de trabalho

OAWS IoTO serviço de trabalhos do publica em tópicos reservados no protocolo MQTT quando os trabalhos estão pendentes, concluídos ou cancelados e quando um dispositivo relata sucesso ou falha ao executar um trabalho. Os dispositivos ou os aplicativos de gerenciamento e monitoramento podem acompanhar o status de trabalhos assinando esses tópicos.

Como habilitar eventos de empregos

Mensagens de resposta doAWS IoTO serviço de empregos não passa pelo corretor de mensagens e eles não podem ser inscritos por outros clientes ou regras. Para se inscrever em mensagens relacionadas à atividade de trabalho, use `onotifyenotify-next`Tópicos Para obter informações sobre tópicos de trabalhos, consulteTópicos de trabalhos (p. 109).

Para ser notificado sobre atualizações de trabalhos, ative esses eventos de trabalhos usando oAWS Management Consoleou usando a API ou a CLI do. Para obter mais informações, consulteHabilite eventos para oAWS IoT (p. 1107)

Como funcionam os eventos de empregos

Como cancelamento ou exclusão do trabalho podem levar algum tempo, duas mensagens são enviadas para indicar o início e o fim de uma solicitação. Por exemplo, quando uma solicitação de cancelamento é iniciada, uma mensagem é enviada para o tópico `$aws/events/job/jobID/cancellation_in_progress`. Quando uma solicitação de cancelamento é concluída, uma mensagem é enviada para o tópico `$aws/events/job/jobID/canceled`.

Um processo similar ocorre para uma solicitação de exclusão do trabalho. Os aplicativos de gerenciamento e de monitoramento podem assinar esses tópicos para acompanhar os status de trabalhos. Para obter

mais informações sobre publicação e assinatura em tópicos MQTT, consulte the section called “Protocolos de comunicação de” (p. 83).

Tipos de evento de Job

A seguir, mostra os diferentes tipos de eventos de trabalho:

Trabalho concluído/cancelado/excluído

O serviço de trabalhos do AWS IoT publica uma mensagem em um tópico MQTT quando um trabalho é concluído, cancelado, excluído ou quando o cancelamento ou a exclusão está em andamento:

- `$aws/events/job/jobID/completed`
- `$aws/events/job/jobID/canceled`
- `$aws/events/job/jobID/deleted`
- `$aws/events/job/jobID/cancellation_in_progress`
- `$aws/events/job/jobID/deletion_in_progress`

A mensagem `completed` contém as seguintes cargas de exemplo:

```
{
  "eventType": "JOB",
  "eventId": "7364ffd1-8b65-4824-85d5-6c14686c97c6",
  "timestamp": 1234567890,
  "operation": "completed",
  "jobId": "27450507-bf6f-4012-92af-bb8a1c8c4484",
  "status": "COMPLETED",
  "targetSelection": "SNAPSHOT|CONTINUOUS",
  "targets": [
    "arn:aws:iot:us-east-1:123456789012:thing/a39f6f91-70cf-4bd2-a381-9c66df1a80d0",
    "arn:aws:iot:us-east-1:123456789012:thinggroup/2fc4c0a4-6e45-4525-
a238-0fe8d3dd21bb"
  ],
  "description": "My Job Description",
  "completedAt": 1234567890123,
  "createdAt": 1234567890123,
  "lastUpdatedAt": 1234567890123,
  "jobProcessDetails": {
    "numberOfCanceledThings": 0,
    "numberOfRejectedThings": 0,
    "numberOfFailedThings": 0,
    "numberOfRemovedThings": 0,
    "numberOfSucceededThings": 3
  }
}
```

O`canceled`A mensagem contém as seguintes cargas de exemplo.

```
{
  "eventType": "JOB",
  "eventId": "568d2ade-2e9c-46e6-a115-18afa1286b06",
  "timestamp": 1234567890,
  "operation": "canceled",
  "jobId": "4d2a531a-da2e-47bb-8b9e-ff5adcd53ef0",
  "status": "CANCELED",
  "targetSelection": "SNAPSHOT|CONTINUOUS",
  "targets": [
    "arn:aws:iot:us-east-1:123456789012:thing/Thing0-947b9c0c-ff10-4a80-b4b3-
cd33d0145a0f",
    "arn:aws:iot:us-east-1:123456789012:thinggroup/ThingGroup1-95c644d5-1621-41a6-9aa5-
ad2de581d18f"
  ],
```

```
  "description": "My job description",
  "createdAt": 1234567890123,
  "lastUpdatedAt": 1234567890123
}
```

O`deleted`A mensagem contém as seguintes cargas de exemplo.

```
{
      "eventType": "JOB",
      "eventId": "568d2ade-2e9c-46e6-a115-18afa1286b06",
      "timestamp": 1234567890,
      "operation": "deleted",
      "jobId": "4d2a531a-da2e-47bb-8b9e-ff5adcd53ef0",
      "status": "DELETED",
      "targetSelection": "SNAPSHOT|CONTINUOUS",
      "targets": [
        "arn:aws:iot:us-east-1:123456789012:thing/Thing0-947b9c0c-ff10-4a80-b4b3-
cd33d0145a0f",
        "arn:aws:iot:us-east-1:123456789012:thinggroup/
ThingGroup1-95c644d5-1621-41a6-9aa5-ad2de581d18f"
      ],
      "description": "My job description",
      "createdAt": 1234567890123,
      "lastUpdatedAt": 1234567890123,
      "comment": "Comment for this operation"
    }
```

A mensagem `cancellation_in_progress` contém as seguintes cargas de exemplo:

```
{
      "eventType": "JOB",
      "eventId": "568d2ade-2e9c-46e6-a115-18afa1286b06",
      "timestamp": 1234567890,
      "operation": "cancellation_in_progress",
      "jobId": "4d2a531a-da2e-47bb-8b9e-ff5adcd53ef0",
      "status": "CANCELLATION_IN_PROGRESS",
      "targetSelection": "SNAPSHOT|CONTINUOUS",
      "targets": [
        "arn:aws:iot:us-east-1:123456789012:thing/Thing0-947b9c0c-ff10-4a80-b4b3-
cd33d0145a0f",
        "arn:aws:iot:us-east-1:123456789012:thinggroup/
ThingGroup1-95c644d5-1621-41a6-9aa5-ad2de581d18f"
      ],
      "description": "My job description",
      "createdAt": 1234567890123,
      "lastUpdatedAt": 1234567890123,
      "comment": "Comment for this operation"
    }
```

A mensagem `deletion_in_progress` contém as seguintes cargas de exemplo:

```
{
      "eventType": "JOB",
      "eventId": "568d2ade-2e9c-46e6-a115-18afa1286b06",
      "timestamp": 1234567890,
      "operation": "deletion_in_progress",
      "jobId": "4d2a531a-da2e-47bb-8b9e-ff5adcd53ef0",
      "status": "DELETION_IN_PROGRESS",
      "targetSelection": "SNAPSHOT|CONTINUOUS",
      "targets": [
        "arn:aws:iot:us-east-1:123456789012:thing/Thing0-947b9c0c-ff10-4a80-b4b3-
cd33d0145a0f",
```

```
        "arn:aws:iot:us-east-1:123456789012:thinggroup/
ThingGroup1-95c644d5-1621-41a6-9aa5-ad2de581d18f"
      ],
      "description": "My job description",
      "createdAt": 1234567890123,
      "lastUpdatedAt": 1234567890123,
      "comment": "Comment for this operation"
    }
```

Status terminal da execução do trabalho

O serviço Trabalhos do AWS IoT publica uma mensagem quando um dispositivo atualiza um status terminal da execução de um trabalho:

- `$aws/events/jobExecution/jobID/succeeded`
- `$aws/events/jobExecution/jobID/failed`
- `$aws/events/jobExecution/jobID/rejected`
- `$aws/events/jobExecution/jobID/canceled`
- `$aws/events/jobExecution/jobID/timed_out`
- `$aws/events/jobExecution/jobID/removed`
- `$aws/events/jobExecution/jobID/deleted`

A mensagem contém as seguintes cargas de exemplo:

```
{
  "eventType": "JOB_EXECUTION",
  "eventId": "cca89fa5-8a7f-4ced-8c20-5e653afb3572",
  "timestamp": 1234567890,
  "operation": "succeeded|failed|rejected|canceled|removed|timed_out",
  "jobId": "154b39e5-60b0-48a4-9b73-f6f8dd032d27",
  "thingArn": "arn:aws:iot:us-east-1:123456789012:myThing/6d639fbc-8f85-4a90-924d-
a2867f8366a7",
  "status": "SUCCEEDED|FAILED|REJECTED|CANCELED|REMOVED|TIMED_OUT",
  "statusDetails": {
    "key": "value"
  }
}
```

Eventos de ciclo de vida

AWS IoTPode publicar eventos de ciclo de vida nos tópicos MQTT. Esses eventos estão disponíveis por padrão e eles não podem ser desabilitados.

> **Note**
>
> As mensagens de ciclo de vida podem ser enviadas fora de ordem. Você pode receber mensagens duplicadas.

Eventos de conexão/desconexão

A AWS IoT publica uma mensagem nos tópicos MQTT a seguir quando um cliente se conecta ou se desconecta:

- `$aws/events/presence/connected/clientId`— um cliente se conectou apenas ao agente de mensagens.

- `$aws/events/presence/disconnected/clientId`— um cliente se desconectou do agente de mensagens.

Veja a seguir uma lista de elementos JSON contidos nas mensagens de conexão/desconexão publicadas no tópico `$aws/events/presence/connected/clientId`.

clientId

O ID do cliente que se conecta ou que se desconecta.

> **Note**
>
> Os IDs de clientes que contêm # ou + não recebem eventos de ciclo de vida.

clientInitiatedDisconnect

Verdadeiro se o cliente iniciou a desconexão. Caso contrário, falso. Encontrado somente em mensagens de desconexão.

disconnectReason

A razão pela qual o cliente está se desconectando. Encontrado apenas em mensagens de desconexão. A tabela a seguir contém valores válidos.

Motivo da desconexão	Descrição
`AUTH_ERROR`	Falha ao autenticar o cliente, ou a autorização falhou.
`CLIENT_INITIATED_DISCONNECT`	O cliente indica que ele se desconectará. O cliente pode fazer isso enviando um MQTT`DISCONNECT`pacote de controle ou um`Close frame`se o cliente estiver usando um WebSocket conexão.
`CLIENT_ERROR`	O cliente fez algo errado que faz com que ele se desconecte. Por exemplo, um cliente será desconectado para enviar mais de 1 pacote `CONNECT` MQTT na mesma conexão ou se o cliente tentar publicar com uma carga útil que exceda o limite de carga útil.
`CONNECTION_LOST`	A conexão cliente-servidor é cortada. Isso pode acontecer durante um período de alta latência de rede ou quando a conexão com a Internet é perdida.
`DUPLICATE_CLIENTID`	O cliente está usando um ID de cliente que já está em uso. Nesse caso, o cliente que já está conectado será desconectado com esse motivo de desconexão.
`FORBIDDEN_ACCESS`	O cliente não tem permissão para ser conectado. Por exemplo, um cliente com um endereço IP negado falhará ao se conectar.
`MQTT_KEEP_ALIVE_TIMEOUT`	Se não houver nenhuma comunicação cliente-servidor para 1,5x do tempo keep-alive do cliente, o cliente será desconectado.

Motivo da desconexão	Descrição
`SERVER_ERROR`	Desconectado devido a problemas inesperados no servidor.
`SERVER_INITIATED_DISCONNECT`	O servidor desconecta intencionalmente um cliente por razões operacionais.
`THROTTLED`	O cliente é desconectado por exceder um limite de controle.
`WEBSOCKET_TTL_EXPIRATION`	O cliente está desconectado porque um WebSocket Tem sido conectado por mais tempo do que seu valor de tempo de vida.

eventType

O tipo de evento. Os valores válidos são `connected` ou `disconnected`.

ipAddress

O endereço IP do cliente que está se conectando. Pode ser no formato IPv4 ou IPv6. Encontrado somente em mensagens de conexão.

principalIdentifier

A credencial usada para autenticar. Para certificados de autenticação mútua TLS, a credencial é o ID do certificado. Para outras conexões, são as credenciais do IAM.

sessionIdentifier

Um identificador globalmente exclusivo na AWS IoT que existe durante o ciclo de vida da sessão.

timestamp

Um aproximação de quando o evento ocorreu, expressa em milissegundos desde o epoch do Unix. A precisão do carimbo de data e hora é de aproximadamente 2 minutos.

versionNumber

O número de versão para o evento de ciclo de vida. Esse é um valor inteiro longo que cresce monotonicamente para cada conexão do ID do cliente. O número da versão pode ser usado por um assinante para inferir a ordem dos eventos de ciclo de vida.

> **Note**
>
> As mensagens de conexão e desconexão para uma conexão do cliente têm o mesmo número de versão.
> O número da versão pode ignorar valores e não há garantia de que ele aumente de forma consistente em 1 para cada evento.
> Se um cliente não permanecer conectado por aproximadamente uma hora, o número da versão será redefinido como 0. Para sessões persistentes, o número da versão é redefinido como 0 depois que um cliente tiver permanecido desconectado por mais tempo do que o configurado time-to-live (TTL) para a sessão persistente.

Uma mensagem de conexão tem a seguinte estrutura.

```
{
   "clientId": "186b5",
   "timestamp": 1573002230757,
```

```
    "eventType": "connected",
    "sessionIdentifier": "a4666d2a7d844ae4ac5d7b38c9cb7967",
    "principalIdentifier": "12345678901234567890123456789012",
    "ipAddress": "192.0.2.0",
    "versionNumber": 0
}
```

Uma mensagem de desconexão tem a seguinte estrutura.

```
{
    "clientId": "186b5",
    "timestamp": 1573002340451,
    "eventType": "disconnected",
    "sessionIdentifier": "a4666d2a7d844ae4ac5d7b38c9cb7967",
    "principalIdentifier": "12345678901234567890123456789012",
    "clientInitiatedDisconnect": true,
    "disconnectReason": "CLIENT_INITIATED_DISCONNECT",
    "versionNumber": 0
}
```

Gerenciamento de desconexões de clientes

A prática recomendada é sempre ter um estado de espera implementado para eventos de ciclo de vida, incluindo mensagens Last Will and Testament (LWT - Último desejo e testamento). Quando uma mensagem de desconexão é recebida, seu código deve aguardar um período e verificar se um dispositivo ainda está offline antes de realizar uma ação. Uma forma de fazer isso é usando Filas de atraso do SQS. Quando um cliente recebe um evento LWT ou de ciclo de vida, você pode colocar uma mensagem na fila (por exemplo, por 5 segundos). Quando essa mensagem se torna disponível e é processada (pelo Lambda ou outro serviço), você pode primeiro verificar se o dispositivo ainda está offline antes de realizar outras ações.

Eventos de assinatura/cancelamento de assinatura

A AWS IoT publica uma mensagem no tópico MQTT a seguir quando um cliente assina ou cancela a assinatura em um tópico MQTT:

```
$aws/events/subscriptions/subscribed/clientId
```

ou

```
$aws/events/subscriptions/unsubscribed/clientId
```

Onde `clientId` é o ID do cliente MQTT que se conecta ao agente de mensagens da AWS IoT.

A mensagem publicada para este tópico tem a seguinte estrutura:

```
{
    "clientId": "186b5",
    "timestamp": 1460065214626,
    "eventType": "subscribed" | "unsubscribed",
    "sessionIdentifier": "00000000-0000-0000-0000-000000000000",
    "principalIdentifier": "000000000000/ABCDEFGHIJKLMNOPQRSTU:some-user/
ABCDEFGHIJKLMNOPQRSTU:some-user",
    "topics" : ["foo/bar","device/data","dog/cat"]
}
```

Veja a seguir uma lista de elementos JSON que estão contidos nas mensagens assinadas e não assinadas publicadas nos tópicos `$aws/events/subscriptions/subscribed/`*`clientId`* e `$aws/events/subscriptions/unsubscribed/`*`clientId`*.

clientId

O ID do cliente que assina ou que cancela a assinatura.

> **Note**
>
> Os IDs de clientes que contêm # ou + não recebem eventos de ciclo de vida.

eventType

O tipo de evento. Os valores válidos são `subscribed` ou `unsubscribed`.

principalIdentifier

A credencial usada para autenticar. Para certificados de autenticação mútua TLS, a credencial é o ID do certificado. Para outras conexões, são as credenciais do IAM.

sessionIdentifier

Um identificador globalmente exclusivo na AWS IoT que existe durante o ciclo de vida da sessão.

timestamp

Um aproximação de quando o evento ocorreu, expressa em milissegundos desde o epoch do Unix. A precisão do carimbo de data e hora é de aproximadamente 2 minutos.

topics

Uma série de tópicos MQTT nos quais o cliente se inscreveu.

> **Note**
>
> As mensagens de ciclo de vida podem ser enviadas fora de ordem. Você pode receber mensagens duplicadas.

AWS IoT Corepara LoRaWAN

AWS IoT Corepara LoRaWAN é um servidor de rede LoRaWAN (LNS) totalmente gerenciado que fornece gerenciamento de gateway usando os recursos do Servidor de Configuração e Atualização (CUPS) e Atualizações de Firmware Over-The-Air (FUOTA). Você pode substituir seu LNS privado porAWS IoT Corepara LoRaWAN e conecte seus dispositivos e gateways de Long Range Wide Area Network (LoRaWAN) paraAWS IoT Core. Ao fazer isso, você reduzirá a manutenção, os custos operacionais, o tempo de configuração e os custos indiretos.

Introdução

Os dispositivos LoRaWAN são dispositivos operados por bateria de longo alcance e baixa potência que usam o protocolo LoRaWAN para operar em um espectro de rádio sem licença. O LoraWAN é um protocolo de comunicação LPWAN (Low Power Wide Area Network) que é construído no LoRa. LoRa é o protocolo de camada física que permite comunicação de baixa potência e ampla área entre dispositivos.

Você pode integrar seus dispositivos LoraWAN da mesma forma que você integraria outros dispositivos IoT paraAWS IoT. Para conectar seus dispositivos LoraWAN aAWS IoT, você deve usar um gateway LorAwan. O gateway atua como uma ponte para conectar seu dispositivo aoAWS IoT Corepara LorAwan e para trocar mensagens.AWS IoT Corepara LoRaWAN usa oAWS IoTmecanismo de regras para rotear as mensagens de seus dispositivos LoraWAN para outrosAWS IoTServiços da .

Para reduzir o esforço de desenvolvimento e integrar rapidamente seus dispositivos paraAWS IoT CorePara a LoraWAN, recomendamos o uso de dispositivos finais certificados pela Lorawan. Para obter mais informações, consulte o .AWS IoT Corepara LoRaWAN. Para obter informações sobre a certificação LoRawan de seus dispositivos, consulteCertificando produtos LoRaWAN.

Como usar oAWS IoT Corepara LoRaWAN

Você pode integrar rapidamente seus dispositivos e gateways LoRawan paraAWS IoT CorePara LorAwan usando o console do ou oAWS IoTAPI Wireless.

Usar o console

Para integrar seus dispositivos e gateways LoRaWAN usando oAWS Management Console, faça login noAWS Management Consolee navegue até oAWS IoT Corepara LoRaWANPágina noAWS IoTconsole do . Você pode usar oIntroseção para adicionar seus gateways e dispositivos aAWS IoT Corepara LoRaWAN. Para obter mais informações, consulte Usando o console para integrar seu dispositivo e gateway paraAWS IoT Core for LoRaWAN (p. 1131).

Usando a API ou a CLI do

Você pode integrar o LorAwan eCalçadaDispositivos usando oAWS IoTSem fioAPI. OAWS IoTAPI Wireless queAWS IoT Corepara LorAwan é construído em é suportado peloAWSSDK. Para obter mais informações, consulteAWSSDKs e kits de ferramentas.

Você pode usar oAWS CLIpara executar comandos para integrar e gerenciar seus gateways e dispositivos Lorawan. Para obter mais informações, consulteAWS IoTReferência de ILC Wireless.

AWS IoT CorePara Regiões e endpoints LoRawan

AWS IoT Corefor LoRaWAN fornece suporte para pontos de extremidade de API de plano de controle e plano de dados específicos para oRegião da AWS. Os endpoints da API do plano de dados são específicos para oConta da AWSeRegião da AWS. Para obter mais informações sobre oAWS IoT Corepara endpoints LoRaWAN, consulteAWS IoT Corepara LoRaWAN EndpointsnoAWSReferência geral.

Para uma comunicação mais segura entre seus dispositivos eAWS IoT, você pode conectar seus dispositivos aAWS IoT Corepara LoRaWANAWS PrivateLinkEm sua Virtual Private Cloud (VPC), em vez de fazer a conexão pela Internet pública. Para obter mais informações, consulte Conexão aoAWS IoT Core for LoRaWANPor meio de um endpoint de interface VPC (p. 1150).

AWS IoT Corepara LoRaWAN tem cotas que se aplicam aos dados do dispositivo que são transmitidos entre os dispositivos e o TPS máximo para oAWS IoTOperações de API Wireless. Para obter mais informações, consulteAWS IoT Corepara cotas LoRaWANnoAWSReferência geral.

AWS IoT Corepara LoRaWAN

Quando você se inscreve paraAWS, você pode começar a usarAWS IoT Corepara LorAwan sem custo usando oAWSNível gratuito.

Para obter mais informações sobre a visão geral do produto e definição de preço do, consulteAWS IoT Corepreços.

O que é oAWS IoT Corepara LoRaWAN?

AWS IoT Corepara LoRaWAN substitui um servidor de rede LoRaWAN (LNS) privado conectando seus dispositivos e gateways LoRawan aAWS. Usar oAWS IoTmecanismo de regras, você pode rotear mensagens recebidas de dispositivos LoraWAN, onde elas podem ser formatadas e enviadas para outrosAWS IoTServiços da . Para proteger as comunicações de dispositivos comAWS IoT,AWS IoT CorePara LoRaWAN usa certificados X.509.

AWS IoT Corefor LorAwan gerencia as políticas de serviço e dispositivo queAWS IoT Corerequer a comunicação com os gateways e dispositivos LoraWAN.AWS IoT Corefor LorAwan também gerencia os destinos que descrevem oAWS IoTregras que enviam dados do dispositivo para outros serviços.

comAWS IoT Corepara LoRaWAN, você pode:

- Embarque e conecte dispositivos e gateways LoRaWAN aAWS IoTSem a necessidade de configurar e gerenciar um LNS privado.
- Connect dispositivos LoRaWAN que estejam em conformidade com as especificações do LoRaWAN 1.0.x ou 1.1 padronizadas por LoRa Aliança. Esses dispositivos podem operar no modo classe A, classe B ou classe C.
- Usar gateways LoraWAN que suportam LoRa Basics Station versão 2.0.4 ou posterior. Todos os gateways qualificados paraAWS IoT Corepara LoRaWAN execute uma versão compatível do LoRa Basics Station.
- Monitore a intensidade do sinal, a largura de banda e o fator de propagação usandoAWS IoT Corepara a taxa de dados adaptativa da LoRaWAN e otimize a taxa de dados, se necessário.
- Atualize o firmware dos gateways LoraWAN usando o serviço CUPS e o firmware dos dispositivos LoRaWAN usando Atualizações de Firmware Over-The-Air (FUOTA).

Tópicos

- O que é LoRaWAN? (p. 1128)
- ComoAWS IoT Corepara LoRaWAN (p. 1129)

O que é LoRaWAN?

OAliança LoRadescreve LorAwan como,“um protocolo de rede Low Power, Wide Area (LPWA) projetado para conectar sem fio 'coisas' operadas por bateria à Internet em redes regionais, nacionais ou globais, e tem como alvo os principais requisitos da Internet das Coisas (IoT), como comunicação bidirecional, end-to-end serviços de segurança, mobilidade e localização.”.

LoRaWAN

O protocolo LoRaWAN é um protocolo de comunicação LPWAN (Low Power Wide Area Networking) que funciona no LoRa. A especificação LoRaWAN está aberta para que qualquer pessoa possa configurar e operar um LoRa Rede.

LoRa é uma tecnologia de frequência de áudio sem fio que opera em um espectro de radiofrequência sem licença. LoRa é um protocolo de camada física que usa modulação de espectro espalhado e suporta comunicação de longo alcance ao custo de uma largura de banda estreita. Ele usa uma forma de onda de banda estreita com uma frequência central para enviar dados, o que o torna robusto à interferência.

Características da tecnologia LoRawan

- Comunicação de longo alcance até 10 milhas na linha de visão.
- Longa duração da bateria de até 10 anos. Para melhorar a duração da bateria, você pode operar seus dispositivos no modo classe A ou classe B, o que requer maior latência de downlink.
- Baixo custo para dispositivos e manutenção.
- Espectro de rádio sem licença, mas regulamentos específicos da região se aplicam.
- Baixo consumo de energia, mas tem um tamanho de carga útil limitado de 51 bytes a 241 bytes, dependendo da taxa de dados. A taxa de dados pode ser de 0,3 Kbit/s — taxa de dados de 27 Kbit/s com um tamanho máximo de carga útil 222.

Saiba mais sobre LoRaWAN

Os links a seguir contêm informações úteis sobre a tecnologia LoRawan e sobre LoRa Basics Station, que é o software que é executado em seus gateways LoRawan para conectar dispositivos finais aAWS IoT Corepara LoRaWAN.

- Os fundamentos das coisas no LorAwan

  The Things Fundamentals on LoRawan contém um vídeo introdutório que abrange os fundamentos do LoRawan e uma série de capítulos que ajudarão você a aprender sobre LoRa e LoRaWAN.
- O que é LoRaWAN

  A LoRa Alliance fornece uma visão geral técnica do LoRa e LorAwan, incluindo um resumo das especificações da LoRaWAN em diferentes regiões.
- LoRa Basics Station

  A Semtech Corporation fornece conceitos úteis sobre LoRa noções básicas para gateways e nós finais. LoRa Basics Station, um software de código aberto executado em seu gateway LoRaWAN, é mantido e distribuído através da Semtech CorporationGitHubrepositório do. Você também pode aprender sobre os protocolos LNS e CUPS que descrevem como trocar dados do LoRaWAN e executar atualizações de configuração.

ComoAWS IoT Corepara LoRaWAN

A arquitetura de rede LoRaWAN está implantada in uma topologia de estrelasem que gateways retransmitem informações entre dispositivos finais e o servidor de rede LoRaWAN (LNS).

AWS IoT Corepara LoRaWAN ajuda você a conectar e gerenciar dispositivos sem fio LoRaWAN (rede de área larga de longo alcance de baixa potência) e substitui a necessidade de você desenvolver e operar um LNS. Dispositivos e gateways WAN de longo alcance (LoRaWAN) podem se conectar aAWS IoT CoreUsando o uso doAWS IoT Corepara LoRaWAN.

A seguir, mostra como um dispositivo LoRaWAN interage comAWS IoT Corepara LoRaWAN. Ela também mostra comoAWS IoT Corepara LorAwan substitui um LNS e se comunica com outrosAWS service (Serviço da AWS)s noNuvem AWS.

Dispositivos LoRaWAN se comunicam comAWS IoT Coreatravés de gateways LoraWAN.AWS IoT Corefor LorAwan gerencia as políticas de serviço e dispositivo queAWS IoT Corerequer gerenciar e se comunicar com os gateways e dispositivos LoraWAN.AWS IoT Corefor LorAwan também gerencia os destinos que descrevem oAWS IoTregras que enviam dados do dispositivo para outros serviços.

Comece a usar oAWS IoT Corepara LoRaWAN

1. Selecione os dispositivos sem fio e os gateways LoRaWAN de que você precisará.

   OAWSCatálogo de dispositivos para parceiroscontém gateways e kits de desenvolvedor qualificados para uso comAWS IoT Corepara LoRaWAN. Para obter mais informações, consulte Usando gateways qualificados a partir doAWSCatálogo de dispositivos para parceiros (p. 1160).
2. Adicione seus dispositivos sem fio e gateways LoraWAN aoAWS IoT Corepara LoRaWAN.

   Conectando gateways e dispositivos aAWS IoT Core for LoRaWAN (p. 1130)fornece informações sobre como descrever seus recursos e adicionar seus dispositivos sem fio e gateways LoraWAN aoAWS IoT Corepara LoRaWAN. Você também aprenderá a configurar o outroAWS IoT Corepara recursos LoraWAN para os quais você precisará gerenciar esses dispositivos e enviar seus dados paraAWSServiços da .
3. Complete seuAWS IoT Corepara LoRaWAN.

   Comece com oNossa amostraAWS IoT Corepara LoRaWANe torne-o seu.

AWS IoT Corepara LoRaWAN

Os seguintes recursos ajudarão você a se familiarizar com a tecnologia LoRaWAN eAWS IoT Corepara LoRaWAN.

- Conceitos básicos doAWS IoT Corepara LoRaWAN

  O vídeo a seguir descreve comoAWS IoT CorePara LorAwan trabalha e orienta você pelo processo de adição de gateways LorAwan doAWS Management Console.

- AWS IoT Corepara LoRaWAN

  O workshop abrange os fundamentos da tecnologia LoRaWAN e sua implementação comAWS IoT Corepara LoRaWAN. Você também pode usar o workshop para percorrer laboratórios que mostram como conectar seu gateway e dispositivo aoAWS IoT Corepara LoraWAN para criar uma solução de IoT de amostra.

Conectando gateways e dispositivos aAWS IoT Core for LoRaWAN

AWS IoT Core for LoRaWANajuda você a conectar e gerenciar dispositivos sem fio LoRaWAN (rede de área larga de longo alcance de baixa potência) e substitui a necessidade de você desenvolver e operar um LNS. Dispositivos e gateways WAN de longo alcance (LoRaWAN) podem se conectar aAWS IoT CoreUsandoAWS IoT Core for LoRaWAN.

Convenções de nomenclatura para seus dispositivos, gateways, perfis e destinos

Antes de começar a usarAWS IoT Core for LoRaWANe criando os recursos, considere a convenção de nomenclatura de seus dispositivos, gateways e destino.

AWS IoT Core for LoRaWANatribui IDs exclusivas aos recursos criados para dispositivos, gateways e perfis sem fio; no entanto, você também pode fornecer aos recursos nomes mais descritivos para facilitar a identificação deles. Antes de adicionar dispositivos, gateways, perfis e destinos aoAWS IoT Core for LoRaWANConsidere como você os nomeará para torná-los mais fáceis de gerenciar.

Você também pode adicionar tags aos recursos que criar. Antes de adicionar seus dispositivos LoraWAN, pense em como você pode usar tags para identificar e gerenciar seusAWS IoT Core for LoRaWANrecursos da AWS. As tags podem ser modificadas após você adicioná-las.

Para obter mais informações sobre nomenclatura e marcação, consulteDescreva suasAWS IoT Corepara recursos LoRaWAN (p. 1131).

Mapeamento de dados do dispositivo para dados de serviço

Os dados dos dispositivos sem fio LoRaWAN geralmente são codificados para otimizar a largura de banda. Essas mensagens codificadas chegam aAWS IoT Core for LoRaWANem um formato que pode não ser facilmente usado por outrosAWSServiços da .AWS IoT Core for LoRaWANusaAWS IoTregras que podem usarAWS Lambdafunções para processar e decodificar as mensagens do dispositivo para um formato que outroAWSserviços podem usar.

Para transformar dados do dispositivo e enviá-los para outrosAWSServiços, você precisa saber:

- O formato e o conteúdo dos dados que os dispositivos sem fio enviam.
- O serviço para o qual você deseja enviar os dados.

- O formato que o serviço requer.

Usando essas informações, você pode criar oAWS IoTregra que executa a conversão e envia os dados convertidos para oAWSserviços que o usarão.

Usando o console para integrar seu dispositivo e gateway paraAWS IoT Core for LoRaWAN

Você pode usar a interface do console ou a API para adicionar seu gateway e dispositivos LoRawan. Se você estiver usandoAWS IoT Core for LoRaWANPela primeira vez, recomendamos que você use o console. A interface do console é mais prática ao gerenciar algunsAWS IoT Core for LoRaWANrecursos de cada vez. Ao gerenciar um grande número deAWS IoT Core for LoRaWANrecursos, considere criar soluções mais automatizadas usando oAWS IoT WirelessAPI.

Grande parte dos dados que você insere ao configurarAWS IoT Core for LoRaWANos recursos são fornecidos pelos fornecedores dos dispositivos e são específicos das especificações da LoRaWAN que eles suportam. Os tópicos a seguir descrevem como você pode descrever oAWS IoT Corepara recursos do LoRaWAN e use o console ou a API para adicionar seus gateways e dispositivos.

Tópicos

- Descreva suasAWS IoT Corepara recursos LoRaWAN (p. 1131)
- Incorporar seus gateways paraAWS IoT Core for LoRaWAN (p. 1133)
- Incorporar seus dispositivosAWS IoT Core for LoRaWAN (p. 1139)

Descreva suasAWS IoT Corepara recursos LoRaWAN

Se você estiver usandoAWS IoT Core for LoRaWANPela primeira vez, você pode adicionar seu primeiro gateway e dispositivo LoRawan usando oAWS IoT Core for LoRaWANPágina de introdução doAWS IoTconsole do .

Antes de começar a criar os recursos, considere a convenção de nomenclatura de seus dispositivos, gateways e destino.AWS IoT Core for LoRaWANfornece várias opções para identificar os recursos que você cria. WHILEAWS IoT Core for LoRaWANos recursos recebem um ID exclusivo quando são criados, esse ID não é descritivo nem pode ser alterado após a criação do recurso. Você também pode atribuir um nome, adicionar uma descrição e anexar tags e valores de tag à maioriaAWS IoT Core for LoRaWANrecursos para tornar mais conveniente selecionar, identificar e gerenciar seuAWS IoT Core for LoRaWANrecursos da AWS.

- Nome de recurso (p. 1132)

  Para gateways, dispositivos e perfis, o nome do recurso é um campo opcional que você pode alterar após a criação do recurso. O nome aparece nas listas exibidas nas páginas do hub de recursos.

  Para destinos, você fornece um nome que é exclusivo em seuAWSConta da eRegião da AWS. Você não pode alterar o nome do destino depois de criar o recurso de destino.

  Embora um nome possa ter até 256 caracteres, o espaço de exibição no hub de recursos é limitado. Certifique-se de que a parte distintiva do nome apareça nos primeiros 20 a 30 caracteres, se possível.
- Tags de recursos (p. 1132)

  As tags são pares de metadados de chave/valor que podem ser anexados aAWSrecursos da AWS. Você escolhe as chaves de tag e seus valores correspondentes.

  Os gateways, destinos e perfis podem ter até 50 tags anexadas a eles. Os dispositivos não oferecem suporte a tags.

Nome de recurso

AWS IoT Core for LoRaWANSuporte de recursos do nome

Recurso	Suporte de campo de nome	
Destination (Destino)	O nome é um ID exclusivo do recurso e não pode ser alterado.	
Dispositivo	O nome é descritor opcional de recurso e pode ser alterado.	
Gateway	O nome é descritor opcional de recurso e pode ser alterado.	
Perfil	O nome é descritor opcional de recurso e pode ser alterado.	

O campo nome aparece nas listas de recursos do hub de recursos; no entanto, o espaço é limitado e, portanto, apenas os primeiros 15-30 caracteres do nome podem estar visíveis.

Ao selecionar nomes para seus recursos, considere como você deseja que eles identifiquem os recursos e como eles serão exibidos no console.

Descrição

Os recursos de destino, dispositivo e gateway também suportam um campo de descrição, que pode aceitar até 2.048 caracteres. O campo de descrição aparece somente na página de detalhes do recurso individual. Embora o campo de descrição possa conter muitas informações, porque ele só aparece na página de detalhes do recurso, não é conveniente para varredura no contexto de vários recursos.

Tags de recursos

AWS IoT Core for LoRaWANSuporte de recursos do para oAWStags

Recurso	AWSSuporte a tags	
Destination (Destino)	Até 50AWSTags podem ser adicionadas ao recurso.	
Dispositivo	Este recurso não oferece suporte aAWStags.	
Gateway	Até 50AWSTags podem ser adicionadas ao recurso.	
Perfil	Até 50AWSTags podem ser adicionadas ao recurso.	

Tags são palavras ou frases que funcionam como metadados que podem ser usados para identificar e organizar oAWSrecursos da AWS. Você pode pensar na chave de tag como uma categoria de informações e o valor da tag como um valor específico nessa categoria.

Por exemplo, você pode ter um valor de tag decore, em seguida, dê a alguns recursos um valor debluepara essa tag e outros um valor devermelho. Com isso, você poderia usar oEditor de tagsnoAWSconsole para encontrar os recursos com umcorValor da tag doblue.

Para obter mais informações sobre estratégias de marcação e marcação, consulteEditor de tags.

Incorporar seus gateways paraAWS IoT Core for LoRaWAN

Se você estiver usandoAWS IoT Core for LoRaWANPela primeira vez, você pode adicionar seu primeiro gateway e dispositivo LoRawan usando o console.

Antes de integrar seu gateway

Antes de integrar seu gateway paraAWS IoT Core for LoRaWAN, recomendamos que você:

- Use gateways qualificados para uso comAWS IoT Core for LoRaWAN. Esses gateways se conectam aAWS IoT Coresem configurações adicionais e tem uma versão compatível do LoRa Basics Stationsoftware em execução neles. Para obter mais informações, consulte Como gerenciar gateways comAWS IoT Corepara LoRaWAN (p. 1159).
- Considere a convenção de nomenclatura dos recursos que você cria para que você possa gerenciá-los com mais facilidade. Para obter mais informações, consulte Descreva suasAWS IoT Corepara recursos LoRaWAN (p. 1131).
- Tenha os parâmetros de configuração exclusivos para cada gateway prontos para entrar com antecedência, o que faz com que a inserção dos dados no console seja mais suave. Os parâmetros de configuração do gateway sem fio queAWS IoTrequer para se comunicar e gerenciar o gateway incluem o EUI do gateway e seu LoRa Faixa de frequência.

Para integrar seus gateways paraAWS IoT Core for LoRaWAN:

- Considere a seleção de banda de frequência e adicione a função do IAM (p. 1133)
- Adicionar um gateway aoAWS IoT Core for LoRaWAN (p. 1135)
- Connect seu gateway LoRaWAN e verifique o status da conexão (p. 1137)

Considere a seleção de banda de frequência e adicione a função do IAM

Antes de adicionar seu gateway aoAWS IoT Core for LoRaWAN, recomendamos que você considere a faixa de frequência na qual seu gateway estará operando e adicione a função do IAM necessária para conectar seu gateway aoAWS IoT Core for LoRaWAN.

> Note
>
> Se estiver adicionando o gateway usando o console, clique emCriar funçãono console para criar a função do IAM necessária para que você possa pular essas etapas. Você precisa executar essas etapas somente se estiver usando a CLI para criar o gateway.

Considere a seleção de LoRa bandas de frequência para seus gateways e conexão de dispositivos

AWS IoT Core for LoRaWANsuporta as bandas de frequência EU863-870, US902-928, AU915 e AS923-1, que você pode usar para conectar seus gateways e dispositivos que estão fisicamente presentes em países que suportam as faixas de frequência e características dessas bandas. As bandas EU863-870 e US902-928 são comumente usadas na Europa e na América do Norte, respectivamente. A banda AS923-1 é comumente usada na Austrália, Nova Zelândia, Japão e Cingapura, entre outros países. O AU915 é usado na Austrália e na Argentina, entre outros países. Para obter mais informações sobre qual faixa de frequência usar na sua região ou país, consulteParâmetros regionais LoRaWAN®.

A LoRa Alliance publica especificações da LoRaWAN e documentos de parâmetros regionais que estão disponíveis para download no LoRa Site da Aliança. O LoRa Os parâmetros regionais da Aliança ajudam as empresas a decidir qual faixa de frequência usar em sua região ou país.AWS IoT Core for LoRaWANa implementação da banda de frequência segue a recomendação no documento de especificação de parâmetros regionais. Esses parâmetros regionais são agrupados em um conjunto de parâmetros de rádio, juntamente com uma alocação de frequência adaptada à banda Industrial, Scientific e Medical (ISM). Recomendamos que você trabalhe com as equipes de conformidade para garantir que você atenda a todos os requisitos regulamentares aplicáveis.

Adicione uma função do IAM para permitir que o Servidor de Configuração e Atualização (CUPS) gerencie credenciais de gateway

Este procedimento descreve como adicionar uma função do IAM que permitirá que o Servidor de Configuração e Atualização (CUPS) gerencie credenciais de gateway. Certifique-se de executar esse procedimento antes que um gateway LoraWAN tente se conectar comAWS IoT Core for LoRaWANNo entanto, você só precisa fazer isso uma vez.

Adicione a função do IAM para permitir que o Servidor de Configuração e Atualização (CUPS) gerencie credenciais de gateway

1. Abrir oHub de funções do console do IAMe escolhaCriar função.
2. Se você acha que já pode ter adicionado oIOTWirelessGatewaycertManagErrolefunção, na barra de pesquisa, insira**`IoTWirelessGatewayCertManagerRole`**.

   Se você vir umIOTWirelessGatewaycertManagErrolenos resultados da pesquisa, você tem a função do IAM necessária. Você pode deixar o procedimento agora.

   Se os resultados da pesquisa estiverem vazios, você não terá a função do IAM necessária. Continue o procedimento para adicioná-lo.
3. DentroSelecione o tipo de entidade confiável, escolhaOutroConta da AWS.
4. DentroID da conta, insira seuConta da AWSID e, em seguida, escolhaPróximo: Permissions
5. Na caixa de pesquisa, insira **`AWSIoTWirelessGatewayCertManager`**.
6. Na lista de resultados da pesquisa, selecione a política chamadaAWSIoTWirelessGatewayCertManager.
7. Selecione Next (Próximo): Tagse, depois, escolhaPróximo: Análise.
8. DentroNome da função, insira**`IoTWirelessGatewayCertManagerRole`**e, depois, escolhaCriar função.
9. Para editar a nova função, na mensagem de confirmação, escolhaIOTWirelessGatewaycertManagErrole.
10. DentroSummary (Resumo), escolha oRelações de confiançae, depois, escolhaEditar relação de confiança.
11. DentroDocumento da política, altere o`Principal`Propriedade para ser semelhante a este exemplo.

```
"Principal": {
    "Service": "iotwireless.amazonaws.com"
},
```

Depois de alterar o`Principal`Propriedade, o documento de política completo deve ser semelhante a este exemplo.

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

```
      "Effect": "Allow",
      "Principal": {
        "Service": "iotwireless.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {}
    }
  ]
}
```

12. Para salvar as alterações e sair, escolhaUpdate Trust Policy.

Agora você criou oIOTWirelessGatewaycertManagErrole. Você não precisará fazer isso de novo.

Se você executou esse procedimento enquanto estava adicionando um gateway, você pode fechar essa janela e o console do IAM e retornar aoAWS IoTconsole para terminar de adicionar o gateway.

Adicionar um gateway aoAWS IoT Core for LoRaWAN

Você pode adicionar seu gateway aoAWS IoT Core for LoRaWANUsando o console do ou a CLI da.

Antes de adicionar seu gateway, recomendamos que você considere os fatores mencionados naAntes de integrar seu gatewayseção doIncorporar seus gateways paraAWS IoT Core for LoRaWAN (p. 1133).

Se estiver adicionando seu gateway pela primeira vez, recomendamos usar o console. Se você quiser adicionar seu gateway usando a CLI, você já deve ter criado a função do IAM necessária para que o gateway possa se conectar comAWS IoT Core for LoRaWAN. Para obter informações sobre como criar a função, consulteAdicione uma função do IAM para permitir que o Servidor de Configuração e Atualização (CUPS) gerencie credenciais de gateway (p. 1134).

Adicionar um gateway usando o console

Navegue até o .AWS IoT Core for LoRaWAN IntroA página doAWS IoTConsole do e escolhaConceitos básicose, depois, escolhaAdicionar gateway. Se você já adicionou um gateway, escolhaExibir gatewaypara exibir o gateway que você adicionou. Se você quiser adicionar mais gateways, escolhaAdicionar gateway.

1. Fornecer detalhes do gateway e informações sobre a faixa de frequência

   Usar aDetalhes do gatewaypara fornecer informações sobre os dados de configuração do dispositivo, como o EUI do Gateway e a configuração da banda de frequência.

   - IUE da Gateway

     O EUI (Extended Unique Identifier) do dispositivo gateway individual. O EUI é um código alfanumérico de 16 dígitos, como`c0ee40ffff29df10`, que identifica exclusivamente um gateway em sua rede LoRawan. Essas informações são específicas do modelo de gateway e você pode encontrá-las no dispositivo gateway ou no manual do usuário.

     > **Note**
     >
     > O EUI do Gateway é diferente do endereço MAC Wi-Fi que você pode ver impresso em seu dispositivo gateway. O EUI segue uma norma EUI-64 que identifica exclusivamente a sua porta de entrada e, portanto, não pode ser retomada em outrosConta da AWSs e regiões.

   - Banda de frequência (RFRegion)

     A banda de frequência do gateway. Você pode escolher entre`US915`,`EU868`,`AU915`, ou`AS923-1`, dependendo do suporte do gateway e de qual país ou região o gateway está se conectando fisicamente. Para obter mais informações sobre as bandas, consulteConsidere a seleção de LoRa bandas de frequência para seus gateways e conexão de dispositivos (p. 1133).

2. Especifique os dados de configuração do gateway sem fio (opcional)

   Esses campos são opcionais e você pode usá-los para fornecer informações adicionais sobre o gateway e sua configuração.

   - Nome, descrição e tags para seu gateway

     As informações nesses campos opcionais vêm de como você organiza e descreve os elementos em seu sistema sem fio. Você pode atribuir umName (Nome)para o gateway, use oDescriçãocampo para fornecer informações sobre o gateway e usarTagspara adicionar pares de metadados de valor chave sobre o gateway. Para obter mais informações sobre como nomear e descrever recursos do, consulteDescreva suasAWS IoT Corepara recursos LoRaWAN (p. 1131).

   - Configuração do LoRaWAN usando subbandas e filtros

     Opcionalmente, você também pode especificar dados de configuração do LoRaWAN, como as subbandas que você deseja usar e filtros que podem controlar o fluxo de tráfego. Para este tutorial, você pode pular esses campos. Para obter mais informações, consulte Configure as subbandas e os recursos de filtragem do gateway (p. 1160).

3. Associar umAWS IoTcoisa com o gateway

   Especifique se deseja criar umAWS IoTE associe-o ao gateway. Coisas emAWS IoTOs dispositivos podem facilitar a pesquisa e o gerenciamento de dispositivos. Associar uma coisa ao seu gateway permite que o gateway acesse outrosAWS IoT Corerecursos.

4. Criar e baixar o certificado de gateway

   Para autenticar seu gateway para que ele possa se comunicar com segurança comAWS IoT, seu gateway LorAwan deve apresentar uma chave privada e um certificado paraAWS IoT Core for LoRaWAN. Criar umCertificado de gatewayde modo queAWS IoTpode verificar a identidade do gateway usando o padrão X.509.

   Clique no ícone da barra de ferramentasCriar certificadoe baixe os arquivos de certificado. Você os usará mais tarde para configurar seu gateway.

5. Copie os endpoints CUPS e LNS e baixe certificados

   Seu gateway LoRaWAN deve se conectar a um endpoint CUPS ou LNS ao estabelecer uma conexão comAWS IoT Core for LoRaWAN. Recomendamos usar o endpoint CUPS, pois ele também fornece gerenciamento de configuração. Para verificar a autenticidade doAWS IoT Core for LoRaWANendpoints, seu gateway usará um certificado de confiança para cada um dos endpoints CUPS e LNS,

   Clique no ícone da barra de ferramentasCopiarpara copiar os endpoints CUPS e LNS. Você precisará dessas informações posteriormente para configurar seu gateway. Em seguida, clique na guiaBaixar certificados de confiança do servidorpara baixar os certificados de confiança para os endpoints CUPS e LNS.

6. Criar a função do IAM para as permissões do gateway

   Você precisa adicionar uma função do IAM que permita que o Servidor de Configuração e Atualização (CUPS) gerencie credenciais de gateway. Você deve fazer isso antes que um gateway LoraWAN tente se conectar comAWS IoT Core for LoRaWAN; no entanto, você precisa fazer isso apenas uma vez.

   Para criar oIOTWirelessGatewayCertManagerFunção do IAM para sua conta, clique noCriar função. Se a função já existir, selecione-a na lista suspensa.

   Clique emEnviarpara concluir a criação do gateway.

Adicione um gateway usando a API

Se você estiver adicionando um gateway pela primeira vez usando a API ou a CLI, você deve adicionar oIOTWirelessGatewayCertManagerFunção do IAM para que o gateway possa se conectar comAWS IoT

Core for LoRaWAN. Para obter informações sobre como criar a função, consulte a seção a seguirAdicione uma função do IAM para permitir que o Servidor de Configuração e Atualização (CUPS) gerencie credenciais de gateway (p. 1134).

As listas a seguir descrevem as ações da API que executam as tarefas associadas à adição, atualização ou exclusão de um gateway LoRaWAN.

AWS IoT WirelessAções da API doAWS IoT Core for LoRaWANgateways

- CreateWirelessGateway
- GetWirelessGateway
- ListWirelessGateways
- UpdateWirelessGateway
- DeleteWirelessGateway

Para a lista completa de ações e tipos de dados disponíveis para criar e gerenciarAWS IoT Core for LoRaWANRecursos, consulte a barra de ferramentasAWS IoT WirelessReferência da API do.

Como usar a barra de ferramentasAWS CLIpara adicionar um gateway

Você pode usar oAWS CLIpara criar um gateway sem fio usando ocreate-wireless-gatewaycomando. O exemplo a seguir cria um gateway de dispositivo LoRaWAN sem fio. Você também pode fornecer um`input.json`arquivo que conterá detalhes adicionais, como o certificado do gateway e as credenciais de provisionamento.

> **Note**
>
> Também é possível executar esse procedimento com a API usando os métodos na API da AWS que correspondam aos comandos da CLI mostrados aqui.

```
aws iotwireless create-wireless-gateway \
    --lorawan GatewayEui="a1b2c3d4567890ab",RfRegion="US915" \
    --name "myFirstLoRaWANGateway" \
    --description "Using my first LoRaWAN gateway"
    --cli-input-json input.json
```

Para obter informações sobre as CLIs que podem ser usadas, consulteAWS CLIreferência

Connect seu gateway LoRaWAN e verifique o status da conexão

Antes de verificar o status da conexão do gateway, você já deve ter adicionado seu gateway e conectado aoAWS IoT Core for LoRaWAN. Para obter informações sobre como adicionar o gateway, consulteAdicionar um gateway aoAWS IoT Core for LoRaWAN (p. 1135).

Connect o gateway aoAWS IoT Core for LoRaWAN

Depois de adicionar o gateway, conecte-se à interface de configuração do gateway para inserir as informações de configuração e os certificados de confiança.

Depois de adicionar as informações do gateway aoAWS IoT Core for LoRaWAN, adicione algunsAWS IoT Core for LoRaWANInformações do dispositivo de gateway. A documentação fornecida pelo fornecedor do gateway deve descrever o processo de upload dos arquivos de certificado para o gateway e configurar o dispositivo gateway para se comunicar comAWS IoT Core for LoRaWAN.

Gateways qualificados para uso comAWS IoT Core for LoRaWAN

Para obter instruções sobre como configurar seu gateway LoRawan, consulte oConfigurar dispositivo de gatewaySeção do doAWS IoT Core for LoRaWANWorkshop. Aqui, você encontrará informações sobre instruções para conectar gateways qualificados para uso comAWS IoT Core for LoRaWAN.

Gateways compatíveis com o protocolo CUPS

As instruções a seguir mostram como você pode conectar seus gateways compatíveis com o protocolo CUPS.

1. Carregue os seguintes arquivos que você obteve ao adicionar seu gateway.
   - Certificado de dispositivo de gateway e arquivos de chave privada.
   - Arquivo de certificado de confiança para endpoint CUPS,`cups.trust`.
2. Especifique o URL do endpoint do CUPS que você obteve anteriormente. O endpoint será do formato`prefix.cups.lorawan.region.amazonaws.com:443`.

Para obter detalhes sobre como obter essas informações, consulteAdicionar um gateway aoAWS IoT Core for LoRaWAN (p. 1135).

Gateways compatíveis com o protocolo LNS

As instruções a seguir mostram como você pode conectar seus gateways compatíveis com o protocolo LNS.

1. Carregue os seguintes arquivos que você obteve ao adicionar seu gateway.
   - Certificado de dispositivo de gateway e arquivos de chave privada.
   - Arquivo de certificado de confiança para endpoint LNS,`lns.trust`.
2. Especifique o URL do endpoint LNS que você obteve anteriormente. O endpoint será do formato`prefix.lns.lorawan.region.amazonaws.com:443`.

Para obter detalhes sobre como obter essas informações, consulteAdicionar um gateway aoAWS IoT Core for LoRaWAN (p. 1135).

Depois disso, você conectou seu gateway aoAWS IoT Core for LoRaWANVocê pode verificar o status da sua conexão e obter informações sobre quando o último uplink foi recebido usando o console do ou a API.

Verifique o status da conexão do gateway usando o console

Para verificar o status da conexão usando o console, navegue até oGateways doA página doAWS IoTconsole e escolha o gateway que você adicionou. NoDetalhes específicos do LoRaWANda página de detalhes do Gateway, você verá o status da conexão e a data e hora em que o último uplink foi recebido.

Verifique o status da conexão do gateway usando a API

Para verificar o status da conexão usando a API, use o`GetWirelessGatewayStatisticsAPI`. Essa API não tem um corpo de solicitação e contém apenas um corpo de resposta que mostra se o gateway está conectado e quando o último uplink foi recebido.

```
HTTP/1.1 200
Content-type: application/json

{
  "ConnectionStatus": "Connected",
  "LastUplinkReceivedAt": "2021-03-24T23:13:08.476015749Z",
  "WirelessGatewayId": "30cbdcf3-86de-4291-bfab-5bfa2b12bad5"
}
```

Incorporar seus dispositivosAWS IoT Core for LoRaWAN

Depois de integrar seu gateway paraAWS IoT Core for LoRaWANe verificou o status da conexão, você pode integrar seus dispositivos sem fio. Para obter informações sobre como integrar seus gateways, consulteIncorporar seus gateways paraAWS IoT Core for LoRaWAN (p. 1133).

Os dispositivos LoRaWAN usam um protocolo LoRaWAN para trocar dados com aplicativos hospedados em nuvem.AWS IoT Core for LoRaWANsuporta dispositivos que atendem às especificações 1.0.x ou 1.1 LoRaWAN padronizadas por LoRa Aliança.

Um dispositivo LoRawan geralmente contém um ou mais sensoreseatores. Os dispositivos enviam dados de telemetria de uplink através de gateways LoRaWAN paraAWS IoT Core for LoRaWAN. Aplicativos hospedados na nuvem podem controlar os sensores enviando comandos de downlink para dispositivos LoRaWAN por meio de gateways LoRaWAN.

Antes de integrar seu dispositivo sem fio

Antes de integrar seu dispositivo sem fio paraAWS IoT Core for LoRaWANVocê precisa ter as seguintes informações prontas com antecedência:

- Especificação LoRaWAN e configuração de dispositivos sem fio

  Ter os parâmetros de configuração exclusivos para cada dispositivo prontos para entrar com antecedência faz com que a inserção dos dados no console seja mais suave. Os parâmetros específicos que você precisa inserir dependem da especificação LoRaWAN que o dispositivo usa. Para obter a lista completa de suas especificações e parâmetros de configuração, consulte a documentação de cada dispositivo.
- Nome e descrição do dispositivo (opcional)

  As informações nesses campos opcionais vêm de como você organiza e descreve os elementos em seu sistema sem fio. Para obter mais informações sobre como nomear e descrever recursos do, consulteDescreva suasAWS IoT Corepara recursos LoRaWAN (p. 1131).
- Perfis de serviço e dispositivo

  Tenhaalguns parâmetros de configuração de dispositivos sem fio prontos que são compartilhados por muitos dispositivos e podem ser armazenados emAWS IoT Core for LoRaWANcomo perfis de dispositivo e serviço. Os parâmetros de configuração são encontrados na documentação do dispositivo ou no próprio dispositivo. Você desejará identificar um perfil de dispositivo que corresponda aos parâmetros de configuração do dispositivo ou criar um, se necessário, antes de adicionar o dispositivo. Para obter mais informações, consulte Adicionar perfis aoAWS IoT Core for LoRaWAN (p. 1141).
- AWS IoT Core for LoRaWANdestino

  Cada dispositivo deve ser atribuído a um destino que processará suas mensagens para enviar paraAWS IoTe outros serviços. OAWS IoTas regras que processam e enviam as mensagens do dispositivo são específicas para o formato de mensagem do dispositivo. Para processar as mensagens do dispositivo e enviá-las para o serviço correto, identifique o destino que você criará para usar com as mensagens do dispositivo e atribua-o ao dispositivo.

Para integrar seu dispositivo sem fio paraAWS IoT Core for LoRaWAN

- Adicione seu dispositivo sem fio aoAWS IoT Core for LoRaWAN (p. 1140)
- Adicionar perfis aoAWS IoT Core for LoRaWAN (p. 1141)
- Adicionar destinos aAWS IoT Corepara LoRaWAN (p. 1144)
- Criar regras para processar mensagens de dispositivo LoraWAN (p. 1147)
- Connect seu dispositivo LoraWAN e verifique o status da conexão (p. 1149)

Adicione seu dispositivo sem fio aoAWS IoT Core for LoRaWAN

Se estiver adicionando seu dispositivo sem fio pela primeira vez, recomendamos usar o console. Navegue até o .AWS IoT Core for LoRaWAN IntroA página doAWS IoTConsole do, escolhaConceitos básicose, depois, escolhaAdicionar dispositivo. Se você já adicionou um dispositivo, escolhaVisualizar dispositivopara exibir o gateway que você adicionou. Se quiser adicionar mais dispositivos, escolhaAdicionar dispositivo.

Como alternativa, você também pode adicionar dispositivos sem fio doDispositivosA página doAWS IoTconsole do .

Adicione a especificação do dispositivo sem fio aoAWS IoT Core for LoRaWANUsar o console

Escolha umEspecificação do dispositivo sem fiocom base no seu método de ativação e na versão do LoraWAN. Uma vez selecionados, seus dados são criptografados com uma chave queAWSé dono e gerencia para você.

Modos de ativação OTAA e ABP

Antes que seu dispositivo LoraWAN possa enviar dados de uplink, você deve concluir um processo chamadoativaçãoouProcesso de junção. Para ativar seu dispositivo, você pode usar OTAA (ativação por via aérea) ou ABP (Ativação por personalização).

O ABP não requer um procedimento de junção e usa chaves estáticas. Quando você usa OTAA, seu dispositivo LoRaWAN envia uma solicitação de junção e o Servidor de Rede pode permitir a solicitação. Recomendamos que você use o OTAA para ativar seu dispositivo porque novas chaves de sessão são geradas para cada ativação, o que o torna mais seguro.

LoRaWAN

Quando você usa OTAA, seu dispositivo LoRaWAN e aplicativos hospedados na nuvem compartilham as chaves raiz. Essas chaves raiz dependem se você está usando a versão v1.0.x ou v1.1. O v1.0.x tem apenas uma chave raiz,AppKey(Chave do aplicativo) enquanto a v1.1 tem duas chaves raiz,AppKey(Chave de aplicativo) eNwkKey(Chave de rede). As chaves de sessão são derivadas com base nas chaves raiz para cada ativação. Ambos osNwkKeyeAppKeysão valores hexadecimais de 32 dígitos fornecidos pelo fornecedor sem fio.

EUIs de dispositivos sem fio

Depois de selecionar oEspecificação do dispositivo sem fio, você vê os parâmetros EUI (Extended Unique Identifier) para o dispositivo sem fio exibido no console. Você pode encontrar essas informações na documentação do dispositivo ou do fornecedor sem fio.

- DevEui: valor hexadêmico de 16 dígitos exclusivo do seu dispositivo e encontrado na etiqueta do dispositivo ou na documentação.
- AppEui: valor hexadêmico de 16 dígitos exclusivo do servidor de junção e encontrado na documentação do dispositivo. Na LoRaWAN versão 1.1, oAppEuié chamado comoJoinEui.

Para obter mais informações sobre os identificadores exclusivos, chaves de sessão e chaves raiz, consulte a LoRa Aliançadocumentação.

Adicione a especificação do dispositivo sem fio aoAWS IoT Core for LoRaWANUsando a API

Se você estiver adicionando um dispositivo sem fio usando a API, você deve criar o perfil do dispositivo e o perfil de serviço antes de criar o dispositivo sem fio. Você usará o perfil do dispositivo e o ID do perfil de

serviço ao criar o dispositivo sem fio. Para obter informações sobre como criar esses perfis usando a API, consulteAdicione um perfil de dispositivo usando a API (p. 1142).

As listas a seguir descrevem as ações da API que executam as tarefas associadas à adição, atualização ou exclusão de um perfil de serviço.

AWS IoT WirelessAções de API para perfis de serviço

- CreateWirelessDevice
- GetWirelessDevice
- ListWirelessDevices
- UpdateWirelessDevice
- DeleteWirelessDevice

Para a lista completa de ações e tipos de dados disponíveis para criar e gerenciarAWS IoT Core for LoRaWANRecursos, consulte a barra de ferramentasAWS IoT WirelessReferência da API do.

Como usar a barra de ferramentasAWS CLIPara criar um dispositivo sem fio

Você pode usar oAWS CLIpara criar um dispositivo sem fio usando ocriar-dispositivo sem fiocomando. O exemplo a seguir cria um dispositivo sem fio usando um arquivo.json de entrada para inserir os parâmetros.

> **Note**
>
> Também é possível executar esse procedimento com a API usando os métodos na API da AWS que correspondam aos comandos da CLI mostrados aqui.

Conteúdo da input.json

```
{
    "Description": "My LoRaWAN wireless device"
    "DestinationName": "IoTWirelessDestination"
    "LoRaWAN": {
        "DeviceProfileId": "ab0c23d3-b001-45ef-6a01-2bc3de4f5333",
        "ServiceProfileId": "fe98dc76-cd12-001e-2d34-5550432da100",
        "OtaaV1_1": {
            "AppKey": "3f4ca100e2fc675ea123f4eb12c4a012",
            "JoinEui": "b4c231a359bc2e3d",
            "NwkKey": "01c3f004a2d6efffe32c4eda14bcd2b4"
        },
        "DevEui": "ac12efc654d23fc2"
    },
    "Name": "SampleIoTWirelessThing"
    "Type": LoRaWAN
}
```

Você pode fornecer esse arquivo como entrada para o`create-wireless-device`comando.

```
aws iotwireless create-wireless-device \
    --cli-input-json file://input.json
```

Para obter informações sobre as CLIs que podem ser usadas, consulteAWS CLIreferência

Adicionar perfis aoAWS IoT Core for LoRaWAN

Perfis de dispositivo e serviço podem ser definidos para descrever configurações comuns de dispositivos. Esses perfis descrevem parâmetros de configuração que são compartilhados por dispositivos para facilitar a adição desses dispositivos.AWS IoT Core for LoRaWANsuporta perfis de dispositivos e perfis de serviço.

Os parâmetros de configuração e os valores a serem inseridos nesses perfis são fornecidos pelo fabricante do dispositivo.

Adicionar perfis de dispositivos

Os perfis de dispositivo definem os recursos do dispositivo e os parâmetros de inicialização que o servidor de rede usa para definir o serviço de acesso por rádio LoRaWAN. Inclui seleção de parâmetros como LoRa Faixa de frequência, LoRa versão de parâmetros regionais e versão MAC do dispositivo. Para saber mais sobre as diferentes bandas de frequência, consulteConsidere a seleção de LoRa bandas de frequência para seus gateways e conexão de dispositivos (p. 1133).

Adicione um perfil de dispositivo usando o console

Se você estiver adicionando um dispositivo sem fio usando o console conforme descrito emAdicione a especificação do dispositivo sem fio aoAWS IoT Core for LoRaWANUsar o console (p. 1140), depois de adicionar a especificação do dispositivo sem fio, você pode adicionar o perfil do dispositivo. Como alternativa, você também pode adicionar dispositivos sem fio doPerfisA página doAWS IoTConsole do noLoRaWANGuia.

Você pode escolher entre perfis de dispositivo padrão ou criar um novo perfil de dispositivo. Recomendamos que você use os perfis de dispositivo padrão. Se o aplicativo exigir que você crie um perfil de dispositivo, forneça umNome do perfil do dispositivo, selecione oBanda de frequência (RFregion)que você está usando para o dispositivo e o gateway e mantenha as outras configurações com os valores padrão, a menos que especificado de outra forma na documentação do dispositivo.

Adicione um perfil de dispositivo usando a API

Se você estiver adicionando um dispositivo sem fio usando a API, deverá criar o perfil do dispositivo antes de criar o dispositivo sem fio.

As listas a seguir descrevem as ações da API que executam as tarefas associadas à adição, atualização ou exclusão de um perfil de serviço.

AWS IoT WirelessAções de API para perfis de serviço

- CreateDeviceProfile
- getDeviceProfile
- ListDeviceProfiles
- updateDeviceProfile
- DeleteDeviceProfile

Para a lista completa de ações e tipos de dados disponíveis para criar e gerenciarAWS IoT Core for LoRaWANRecursos, consulte a barra de ferramentasAWS IoT WirelessReferência da API do.

Como usar a barra de ferramentasAWS CLIPara criar um perfil de dispositivo

Você pode usar oAWS CLIPara criar um perfil de dispositivo usando ocreate-device-profilecomando. O exemplo a seguir cria um perfil de dispositivo.

```
aws iotwireless create-device-profile
```

A execução desse comando cria automaticamente um perfil de dispositivo com um ID que você pode usar ao criar o dispositivo sem fio. Agora você pode criar o perfil de serviço usando a seguinte API e, em seguida, criar o dispositivo sem fio usando os perfis de dispositivo e serviço.

```
{
```

```
    "Arn": "arn:aws:iotwireless:us-east-1:123456789012:DeviceProfile/12345678-
a1b2-3c45-67d8-e90fa1b2c34d",
    "Id": "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
}
```

Para obter informações sobre as CLIs que podem ser usadas, consulteAWS CLIreferência

Adicionar perfis de serviço

Os perfis de serviço descrevem os parâmetros de comunicação que o dispositivo precisa para se comunicar com o servidor de aplicativos.

Adicionar um perfil de serviço usando o console

Se você estiver adicionando um dispositivo sem fio usando o console, conforme descrito emAdicione a especificação do dispositivo sem fio aoAWS IoT Core for LoRaWANUsar o console (p. 1140), depois de adicionar o perfil do dispositivo, você pode adicionar seu perfil de serviço. Como alternativa, você também pode adicionar dispositivos sem fio doPerfisA página doAWS IoTConsole do noLoRaWANGuia.

Recomendamos deixar a configuraçãoAddGwMetadatahabilitado para que você receba metadados de gateway adicionais para cada carga útil, como RSSI e SNR para a transmissão de dados.

Adicionar um perfil de serviço usando a API

Se você estiver adicionando um dispositivo sem fio usando a API, primeiro você deve criar seu perfil de serviço antes de criar o dispositivo sem fio.

As listas a seguir descrevem as ações da API que executam as tarefas associadas à adição, atualização ou exclusão de um perfil de serviço.

AWS IoT WirelessAções de API para perfis de serviço

- CreateServiceProfile
- getServiceProfile
- ListServiceProfiles
- updateServiceProfile
- DeleteserviceProfile

Para a lista completa de ações e tipos de dados disponíveis para criar e gerenciarAWS IoT Core for LoRaWANRecursos, consulte a barra de ferramentasAWS IoT WirelessReferência da API do.

Como usar a barra de ferramentasAWS CLIpara criar um perfil de serviço

Você pode usar oAWS CLIpara criar um serviço usando ocreate-service-profilecomando. O exemplo a seguir cria um perfil de serviço.

```
aws iotwireless create-service-profile
```

A execução desse comando cria automaticamente um perfil de serviço com um ID que você pode usar ao criar o dispositivo sem fio. Agora você pode criar o dispositivo sem fio usando os perfis de dispositivo e serviço.

```
{
    "Arn": "arn:aws:iotwireless:us-east-1:123456789012:ServiceProfile/12345678-
a1b2-3c45-67d8-e90fa1b2c34d",
    "Id": "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
}
```

Adicionar destinos aAWS IoT Corepara LoRaWAN

AWS IoT Corepara destinos LoRaWAN descreva oAWS IoTregra que processa os dados de um dispositivo para uso porAWSServiços da .

Porque a maioria dos dispositivos LoRaWAN não envia dados paraAWS IoT Corepara LoRaWAN em um formato que pode ser usado porAWSserviços,AWS IoTregra deve processá-la primeiro. OAWS IoTregra contém a instrução SQL que interpreta os dados do dispositivo e as ações de regra de tópico que enviam o resultado da instrução SQL para os serviços que a usarão.

Se estiver adicionando seu destino pela primeira vez, recomendamos usar o console.

Adicionar um destino usando o console

Se você estiver adicionando um dispositivo sem fio usando o console, conforme descrito emAdicione a especificação do dispositivo sem fio aoAWS IoT Core for LoRaWANUsar o console (p. 1140), depois de você já ter adicionado a especificação e os perfis do dispositivo sem fio aoAWS IoT Core for LoRaWANconforme descrito anteriormente, você pode ir em frente e adicionar um destino.

Como alternativa, você também pode adicionar umAWS IoT Core for LoRaWANDestino doDestinosA página doAWS IoTconsole do .

Para processar os dados de um dispositivo, especifique os seguintes campos ao criar umAWS IoT Corepara o destino LoRaWAN e, em seguida, escolhaAdicionar destino.

- Detalhes do destino

  Digite umNome do destinoe uma descrição opcional para seu destino.
- Nome da regra

  OAWS IoTregra configurada para avaliar mensagens enviadas pelo dispositivo e processar os dados do dispositivo. O nome da regra será mapeado para o seu destino. O destino requer que a regra processe as mensagens recebidas. Você pode escolher as mensagens a serem processadas invocando umAWS IoTregra ou publicando noAWS IoTAgente de mensagens.
  - Se escolherInsira um nome de regra, insira um nome e escolhaCopiarpara copiar o nome da regra que você inserirá ao criar oAWS IoTregra. Você também pode escolherCriar regrapara criar a regra agora ou navegar até oRegrasHub doAWS IoTconsole e crie uma regra com esse nome.

    Você também pode inserir uma regra e usar oAdvanced (Avançado)configuração para especificar um nome de tópico. O nome do tópico é fornecido durante a invocação da regra e é acessado usando `otopic`expressão dentro da regra. Para obter mais informações sobre regras de AWS IoT, consulte Regras para AWS IoT (p. 478).
  - Se escolherPublish to (Publicar no &CW;)AWS IoTAgente de mensagens, insira um nome de tópico. Em seguida, você pode copiar o nome do tópico MQTT e vários assinantes podem se inscrever neste tópico para receber mensagens publicadas nesse tópico. Para obter mais informações, consulte Tópicos MQTT (p. 99).

  Para obter mais informações sobreAWS IoTregras para destinos, consulteCriar regras para processar mensagens de dispositivo LoraWAN (p. 1147).
- Nome da função

  A função do IAM que concede aos dados do dispositivo permissão para acessar a regra nomeada emRule name (Nome da regra). No console, você pode criar uma nova função de serviço ou selecionar uma função de serviço existente. Se você estiver criando uma nova função de serviço, poderá inserir um nome de função (por exemplo,**`IoTWirelessDestinationRole`**), ou deixe em branco paraAWS IoT Core for LoRaWANPara gerar um novo nome de função.AWS IoT Core for LoRaWANEm seguida, criará automaticamente a função do IAM com as devidas permissões em seu nome.

Para obter mais informações sobre funções do IAM, consulteUso de funções do IAM.

Adicione um destino usando a API

Se você deseja adicionar um destino usando a CLI, é necessário já ter criado a regra e a função do IAM para o seu destino. Para obter mais informações sobre os detalhes que um destino requer na função, consulteCriar uma função do IAM para seus destinos (p. 1145).

A lista a seguir contém as ações da API que executam as tarefas associadas à adição, atualização ou exclusão de um destino.

AWS IoT WirelessAções de API do para destinos

- CriadoEstinação
- GetDestination
- ListDestinations
- UpdateDestination
- DeleteDestination

Para a lista completa de ações e tipos de dados disponíveis para criar e gerenciarAWS IoT Core for LoRaWANRecursos, consulte a barra de ferramentasAWS IoT WirelessReferência da API do.

Comopara usar oAWS CLIPara adicionar um destino

Você pode usar oAWS CLIpara adicionar um destino usando oCriar destinocomando. O seguinte exemplo mostra como criar um destino inserindo um nome de regra usando o`RuleName`como o valor para o`expression-type`parâmetro . Se você quiser especificar um nome de tópico para publicar ou se inscrever no corretor de mensagens, altere o`expression-type`Valor do parâmetro para`MqttTopic`d.

```
aws iotwireless create-destination \
    --name IoTWirelessDestination \
    --expression-type RuleName \
    --expression IoTWirelessRule \
    --role-arn arn:aws:iam::123456789012:role/IoTWirelessDestinationRole
```

A execução desse comando cria um destino com o nome de destino, o nome da regra e o nome da função especificados. Para obter informações sobre nomes de regras e funções para destinos, consulteCriar regras para processar mensagens de dispositivo LoraWAN (p. 1147)eCriar uma função do IAM para seus destinos (p. 1145).

Para obter informações sobre as CLIs que podem ser usadas, consulteAWS CLIreferência.

Criar uma função do IAM para seus destinos

AWS IoT Core for LoRaWANdestinos exigem funções do IAM que dãoAWS IoT Core for LoRaWANAs permissões necessárias para enviar dados aoAWS IoTregra. Se essa função ainda não estiver definida, você deve defini-la para que ela apareça na lista de funções.

Quando você usa o console para adicionar um destino,AWS IoT Core for LoRaWANCrie automaticamente uma função do IAM para você, conforme descrito anteriormente neste tópico. Ao adicionar um destino usando a API ou a CLI, é necessário criar a função do IAM para seu destino.

Para criar uma política do IAM para oAWS IoT Core for LoRaWANFunção de destino

1. Abrir oHub de políticas do console do IAM.
2. SelecioneCriar política, e escolha oJSONGuia.

3. No editor, exclua qualquer conteúdo do editor e cole este documento de política.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:DescribeEndpoint",
                "iot:Publish"
            ],
            "Resource": "*"
        }
    ]
}
```

4. SelecionePolítica de revisão, e emName (Nome), insira um nome para essa política. Você precisará desse nome para usar no próximo procedimento.

   Você também pode descrever essa política emDescriçãoSe você quiser.
5. Escolha Create policy (Criar política).

Para criar uma função do IAM para umAWS IoT Core for LoRaWANdestino

1. Abrir oHub de funções do console do IAMe escolhaCriar função.
2. DentroSelecione o tipo de entidade confiável, escolhaOutroConta da AWS.
3. DentroID da conta, insira seuConta da AWSID e, em seguida, escolhaPróximo: Permissions
4. Na caixa de pesquisa, insira o nome da política do IAM criada no procedimento anterior.
5. Nos resultados da pesquisa, verifique a política do IAM criada no procedimento anterior.
6. Selecione Next (Próximo): Tagse, depois, escolhaPróximo: Análise.
7. DentroNome da função, digite o nome dessa função e escolhaCriar função.
8. Na mensagem de confirmação, escolha o nome da função criada para editar a nova função.
9. DentroSummary (Resumo), escolha oRelações de confiançae, depois, escolhaEditar relação de confiança.
10. DentroDocumento da política, altere o`Principal`Propriedade para ser semelhante a este exemplo.

```
"Principal": {
    "Service": "iotwireless.amazonaws.com"
},
```

Depois de alterar o`Principal`Propriedade, o documento de política completo deve ser semelhante a este exemplo.

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

11. Para salvar as alterações e sair, escolhaUpdate Trust Policy.

Com essa função definida, você pode encontrá-la na lista de funções ao configurar oAWS IoT Core for LoRaWANDestinos.

Criar regras para processar mensagens de dispositivo LoraWAN

AWS IoTas regras enviam mensagens de dispositivo para outros serviços.AWS IoTregras também podem processar as mensagens binárias recebidas de um dispositivo LoRaWAN para converter as mensagens em outros formatos que podem facilitar o uso de outros serviços.

AWS IoT Core for LoRaWANdestinos (p. 1144)associe um dispositivo sem fio à regra que processa os dados da mensagem do dispositivo para enviar a outros serviços. A regra atua nos dados do dispositivo assim queAWS IoT Core for LoRaWANRecebe.AWS IoT Core for LoRaWANdestinos (p. 1144)podem ser compartilhados por todos os dispositivos cujas mensagens têm o mesmo formato de dados e que enviam seus dados para o mesmo serviço.

ComoAWS IoTregras processam mensagens do dispositivo

Como umAWS IoTos dados de mensagem de um dispositivo dependem do serviço que receberá os dados, do formato dos dados da mensagem do dispositivo e do formato de dados exigido pelo serviço. Normalmente, a regra chama umAWS Lambdapara converter os dados da mensagem do dispositivo para o formato que um serviço requer e, em seguida, envia o resultado para o serviço.

A ilustração a seguir mostra como os dados da mensagem são protegidos e processados à medida que se movem do dispositivo sem fio para umAWSserviço.

1. O dispositivo sem fio LoRaWAN criptografa suas mensagens binárias usando o modo AES128 CTR antes de transmiti-las.
2. AWS IoT Core for LoRaWANdescriptografa a mensagem binária e codifica a carga útil da mensagem binária descriptografada como uma string base64.
3. A mensagem codificada em base64 resultante é enviada como uma carga de mensagem binária (uma carga útil de mensagem que não está formatada como um documento JSON) para oAWS IoTregra descrita no destino atribuído ao dispositivo.
4. OAWS IoTregra direciona os dados da mensagem para o serviço descrito na configuração da regra.

A carga binária criptografada recebida do dispositivo sem fio não é alterada ou interpretada porAWS IoT Core for LoRaWAN. A carga útil da mensagem binária descriptografada é codificada apenas como uma

string base64. Para que os serviços acessem os elementos de dados na carga da mensagem binária, os elementos de dados devem ser analisados fora da carga por uma função chamada pela regra. A carga útil da mensagem codificada em base64 é uma string ASCII, portanto, ela pode ser armazenada como tal para ser analisada posteriormente.

Criar regras para LoRaWAN

AWS IoT Core for LoRaWANusaAWS IoTregras para enviar mensagens de dispositivo com segurança diretamente para outrosAWSserviços sem a necessidade de usar o corretor de mensagens. Ao remover o corretor de mensagens do caminho de ingestão, ele reduz custos e otimiza o fluxo de dados.

Para umAWS IoT Core for LoRaWANregra para enviar mensagens de dispositivo para outrosAWSserviços, requer umAWS IoT Core for LoRaWANDestino e umAWS IoTregra atribuída a esse destino. OAWS IoTregra deve conter uma instrução de consulta SQL e pelo menos uma ação de regra.

Normalmente, oAWS IoTA instrução de consulta de regra consiste em:

- Uma cláusula SQL SELECT que seleciona e formata os dados da carga útil da mensagem
- Um filtro de tópico (o objeto FROM na instrução de consulta de regra) que identifica as mensagens a serem usadas
- Uma instrução condicional opcional (uma cláusula SQL WHERE) que especifica condições nas quais agir

Veja a seguir um exemplo de uma instrução de consulta de regra:

```
SELECT temperature FROM iot/topic' WHERE temperature > 50
```

Ao construirAWS IoTregras para processar cargas úteis de dispositivos LoraWAN, você não precisa especificar a cláusula FROM como parte do objeto de consulta de regra. A instrução de consulta de regra deve ter a cláusula SQL SELECT e pode opcionalmente ter a cláusula WHERE. Se a instrução query usar a cláusula FROM, ela será ignorada.

Este é um exemplo de uma instrução de consulta de regra que pode processar cargas úteis de dispositivos LoRawan:

```
SELECT WirelessDeviceId, WirelessMetadata.LoRaWAN.FPort as FPort,
       WirelessMetadata.LoRaWAN.DevEui as DevEui,
       PayloadData
```

Neste exemplo, a ação`PayloadData`aé uma carga binária codificada em base64 enviada pelo seu dispositivo LoraWAN.

Aqui está um exemplo de instrução de consulta de regra que pode executar uma decodificação binária da carga de entrada e transformá-la em um formato diferente, como JSON:

```
SELECT WirelessDeviceId, WirelessMetadata.LoRaWAN.FPort as FPort,
       WirelessMetadata.LoRaWAN.DevEui as DevEui,
       aws_lambda("arn:aws:lambda:<region>:<account>:function:<name>",

                  {
                   "PayloadData":PayloadData,
                   "Fport": WirelessMetadata.LoRaWAN.FPort
                  }) as decodingoutput
```

Para obter mais informações sobre como usar as cláusulas SELECT AND WHERE, consulteReferência SQL do AWS IoT (p. 568)

Para obter informações sobre oAWS IoTRegras e como criá-las e usá-las, consulteRegras para AWS IoT (p. 478)eCriarAWS IoTregras para rotear dados do dispositivo para outros serviços (p. 193).

Para obter informações sobre como criar e usarAWS IoT Core for LoRaWANdestinos, consulteAdicionar destinos aAWS IoT Corepara LoRaWAN (p. 1144).

Para obter informações sobre o uso de cargas de mensagens binárias em uma regra, consulteComo trabalhar com cargas úteis binárias (p. 630).

Para obter mais informações sobre a segurança de dados e criptografia usadas para proteger a carga útil da mensagem em sua jornada, consulteProteção de dados no AWS IoT Core (p. 380).

Para uma arquitetura de referência que mostra um exemplo de decodificação e implementação binárias para regras de IoT, consulteAWS IoT Core for LoRaWANExemplos de solução no GitHub.

Connect seu dispositivo LoraWAN e verifique o status da conexão

Antes de verificar o status da conexão do dispositivo, você já deve ter adicionado seu dispositivo e conectado aAWS IoT Core for LoRaWAN. Para obter informações sobre como adicionar o dispositivo, consulteAdicione seu dispositivo sem fio aoAWS IoT Core for LoRaWAN (p. 1140).

Depois de adicionar o dispositivo, consulte o manual do usuário do dispositivo para saber como iniciar o envio de uma mensagem de uplink do seu dispositivo LoRawan.

Verifique o status da conexão do dispositivo usando o console

Para verificar o status da conexão usando o console, navegue até oDispositivosA página doAWS IoTconsole e escolha o dispositivo que você adicionou. NoDetalhesda página de detalhes dos dispositivos sem fio, você verá a data e a hora em que o último uplink foi recebido.

Verifique o status da conexão do dispositivo usando a API

Para verificar o status da conexão usando a API, use o`GetWirelessDeviceStatistics`API. Essa API não tem um corpo de solicitação e contém apenas um corpo de resposta que mostra quando o último uplink foi recebido.

```
HTTP/1.1 200
Content-type: application/json

{
  “LastUplinkReceivedAt”: “2021-03-24T23:13:08.476015749Z”,
  "LoRaWAN": {
        "DataRate": 5,
        "DevEui": "647fda0000006420",
        "Frequency": 868100000
        "Gateways": [
         {
            "GatewayEui": "c0ee40ffff29df10",
            "Rssi": -67,
            "Snr": 9.75
         }
      ],
  “WirelessDeviceId”: “30cbdcf3-86de-4291-bfab-5bfa2b12bad5"
}
```

Next steps (Próximas etapas)

Agora que você conectou seu dispositivo e verificou o status da conexão, você pode observar o formato dos metadados de uplink recebidos do dispositivo usando oCliente de teste MQTTnoTesteA página

doAWS IoTconsole do . Para obter mais informações, consulte Exibir o formato das mensagens de uplink enviadas de dispositivos LoraWAN (p. 1174).

Conexão aoAWS IoT Core for LoRaWANPor meio de um endpoint de interface VPC

Você pode se conectar diretamente aoAWS IoT Core for LoRaWANatravésVPC endpoints de interface (AWS PrivateLink)Em sua Virtual Private Cloud (VPC) em vez de se conectar pela internet pública. Quando você usa um endpoint de interface de VPC, a comunicação entre a VPC e o AWS IoT Core for LoRaWAN é realizada integralmente e com segurança na rede da AWS.

AWS IoT Core for LoRaWANÉ compatível com endpoints de interface do Amazon Virtual Private Cloud que são fornecidos peloAWS PrivateLink. Cada endpoint da VPC é representado por uma ou mais interfaces de rede elástica (ENIs) com endereços IP privados em sua sub-redes da VPC.

Para obter mais informações sobre VPC e endpoints, consulteO que é o Amazon VPC.

Para obter mais informações sobreAWS PrivateLink, consulteAWS PrivateLinke endpoints da VPC.

AWS IoT Core for LoRaWANArquitetura privatelink

O diagrama a seguir mostra a arquitetura privatelink doAWS IoT Core for LoRaWAN. A arquitetura usa um Transit Gateway e Route 53 Resolvedor para compartilhar oAWS PrivateLinkendpoints de interface entre sua VPC, oAWS IoT Core for LoRaWANVPC e um ambiente no local. Você encontrará um diagrama de arquitetura mais detalhado ao configurar a conexão com os endpoints da interface VPC.

Endpoint do AWS IoT Core for LoRaWAN

AWS IoT Core for LoRaWANTem três endpoints públicos. Cada endpoint público tem um endpoint de interface VPC correspondente. Os endpoints públicos podem ser classificados em terminais de plano de

controle e plano de dados. Para obter informações sobre esses endpoints, consulteAWS IoT Core for LoRaWANEndpoints da API.

> Note
>
> AWS PrivateLinkO suporte para os endpoints está disponível apenas no Leste dos EUA (Norte da Virgínia) e Europa (Irlanda).

- Endpoints de API do plano de controle

  Você pode usar os endpoints do plano de controle da API do para interagir com aAWS IoT WirelessAPIs. Esses endpoints podem ser acessados a partir de um cliente hospedado em sua Amazon VPC usandoAWS PrivateLink.
- Endpoints da API do plano de dados

  Os pontos de extremidade da API do plano de dados são LoRaWAN Network Server (LNS) e pontos de extremidade do Servidor de Configuração e Atualização (CUPS) que você pode usar para interagir com oAWS IoT Core for LoRaWANLNS eEndpoints do CUPS. Esses endpoints podem ser acessados a partir de seus gateways LoRa no local usandoAWS VPNouAWS Direct Connect. Você obtém esses endpoints ao integrar seu gateway paraAWS IoT Core for LoRaWAN. Para obter mais informações, consulte Adicionar um gateway aoAWS IoT Core for LoRaWAN (p. 1135).

Os tópicos a seguir mostram como integrar esses endpoints.

Tópicos

- IntegraçãoAWS IoT Core for LoRaWANEndpoint de API do plano de controle (p. 1151)
- IntegraçãoAWS IoT Core for LoRaWANEndpoints de API do plano de dados (p. 1154)

IntegraçãoAWS IoT Core for LoRaWANEndpoint de API do plano de controle

Você pode usarAWS IoT Core for LoRaWANpontos de extremidade da API do plano de controle para interagir com oAWS IoT WirelessAPIs. Por exemplo, é possível usar esse endpoint para executar aSendDataToWirelessDeviceAPI para enviar dados deAWS IoTPara o dispositivo LorAwan. Para obter mais informações, consulteAWS IoT Core for LoRaWANEndpoints de API do plano de controle.

Você pode usar o cliente hospedado em sua Amazon VPC para acessar os endpoints do plano de controle que são alimentados porAWS PrivateLink. Use esses endpoints para se conectar aoAWS IoT WirelessAPI por meio de um endpoint de interface na sua Virtual Private Cloud (VPC) em vez de se conectar pela internet pública.

Para integrar o endpoint do plano de controle:

- Crie sua Amazon VPC e sub-rede (p. 1151)
- Execute uma instância do Amazon EC2 na sub-rede (p. 1152)
- Criar endpoint de interface da Amazon VPC (p. 1152)
- Testar a conexão com o endpoint de interface (p. 1153)

Crie sua Amazon VPC e sub-rede

Antes de poder se conectar ao endpoint da interface, você deve criar uma VPC e uma sub-rede. Em seguida, você executará uma instância do EC2 em sua sub-rede, que pode ser usada para se conectar ao endpoint da interface.

Para criar sua VPC:

1. Navegue até o .VPCsEscolha a página do console da Amazon VPCCriar a VPC.
2. NoCriar a VPC:
   - Insira um nome paraEtiqueta VPC Name - opcional(por exemplo,**`VPC-A`**).
   - Insira um intervalo de endereços IPv4 para sua VPC no bloco CIDR IPv4 (por exemplo,**`10.100.0.0/16`**).
3. Mantenha os valores padrão para outros campos e escolhaCriar a VPC.

Para criar sua sub-rede:

1. Navegue até o .Sub-redesEscolha a página do console da Amazon VPCCriar sub-rede.
2. NoCriar sub-rede:
   - para oID DA VPC, escolha a VPC criada anteriormente (por exemplo,`VPC-A`).
   - Insira um nome paraSubnet name (Nome da sub-rede)(por exemplo,**`Private subnet`**).
   - Selecione oZona de disponibilidadepara a sub-rede.
   - Insira o bloco de endereços IP da sub-rede noIPv4 CIDR block (Bloco CIDR IPv4)no formato CIDR (por exemplo,**`10.100.0.0/24`**).
3. Para criar sua sub-rede e adicioná-la à sua VPC, escolhaCriar sub-rede.

Para obter mais informações, consulteTrabalhar com VPCs e sub-redes.

Execute uma instância do Amazon EC2 na sub-rede

Para iniciar a instância do EC2:

1. Navegue até o .Amazon EC2e selecioneExecução da instância.
2. Para AMI, escolhaAMI do Amazon Linux 2 (HVM), tipo de volume SSDE, em seguida, escolha ot2 microTipo de instância. Para configurar os detalhes da instância, escolhaPróximo.
3. NoConfigurar os detalhes da instância:
   - para oRede, escolha a VPC criada anteriormente (por exemplo,`VPC-A`).
   - para oSubnet (Sub-rede), escolha a sub-rede criada anteriormente (por exemplo,**`Private subnet`**).
   - para oIAM role (Função do IAM), escolha a funçãoAWSIoTWirelessFullAccessconcederAWS IoT Core for LoRaWANpolítica de acesso total. Para obter mais informações, consulte`AWSIoTWirelessFullAccess`Resumo da política.
   - para oSuponha o IP privado, use um endereço IP, por exemplo,10.100.0.42.
4. Selecione Next (Próximo): Add Storagee, depois, escolhaPróximo: Adicionar tags. Opcionalmente, você pode adicionar quaisquer tags para associar à sua instância do EC2. Selecione Next (Próximo): Configurar o grupo de segurança.
5. NoConfigurar o grupo de segurança, configure o security group para permitir:
   - AbertoTodos os TCPpara Source como`10.200.0.0/16`.
   - AbertoTodos ICMP - IPV4para Source como`10.200.0.0/16`.
6. Para revisar os detalhes da instância e executar sua instância do EC2, escolhaReview and Launch.

Para obter mais informações, consulteComece a usar instâncias Linux do Amazon EC2.

Criar endpoint de interface da Amazon VPC

Você pode criar um VPC endpoint para sua VPC, que pode ser acessada pela API do EC2. Para criar o endpoint:

1. Navegue até o .VPC Endpoints doe selecioneCriarPonto final.
2. NoCriar endpoint, especifique as seguintes informações.
   - SelecioneAWS service (Serviço da AWS)speloCategoria de serviço.
   - para oNome do serviço, pesquise digitando a palavra-chave**`iotwireless`**. Na lista de`iotwirelessserviços` exibidos, escolha o endpoint da API do plano de controle para sua Região. O endpoint estará no formato`com.amazonaws.`*`region`*`.iotwireless.api`.
   - para oVPCeSub-redes, escolha a VPC na qual você deseja criar o endpoint e as Zonas de disponibilidade (AZs) nas quais você deseja criar a rede de endpoint.

     Note

     O`iotwirelessO` serviço pode não oferecer suporte a todas as zonas de disponibilidade.
   - para oHabilitar o nome DNS, escolhaAtivar para este endpoint.

     Escolher essa opção resolverá automaticamente o DNS e criará uma rota noAmazon Route 53 Public Data Planepara que as APIs usadas posteriormente para testar a conexão passem pelos endpoints privatelink.
   - para oGrupo de segurança, escolha os grupos de segurança que você deseja associar às interfaces de rede do endpoint.
   - Opcionalmente, você pode adicionar ou remover tags. Tags são pares nome/valor usados para associar ao endpoint.
3. Para criar seu endpoint de VPC, escolhaCriarEndpoint.

Testar a conexão com o endpoint de interface

Você pode usar um SSH para acessar sua instância do Amazon EC2 e, em seguida, usar oAWS CLIpara se conectar aos endpoints da interface privatelink.

Antes de se conectar ao endpoint da interface, baixe o mais recenteAWS CLIversão seguindo as instruções descritas emInstalar, atualizar e desinstalarAWS CLIVersão 2 no Linux.

Os exemplos a seguir mostram como você pode testar sua conexão com o endpoint da interface usando a CLI.

```
aws iotwireless create-service-profile \
    --endpoint-url https://api.iotwireless.region.amazonaws.com  \
    --name='test-privatelink'
```

Veja a seguir um exemplo de execução do comando.

```
Response:
{
 "Arn": "arn:aws:iotwireless:region:acct_number:ServiceProfile/1a2345ba-4c5d-67b0-ab67-
e0c8342f2857",
 "Id": "1a2345ba-4c5d-67b0-ab67-e0c8342f2857"
}
```

Da mesma forma, você pode executar os seguintes comandos para obter as informações do perfil de serviço ou listar todos os perfis de serviço.

```
aws iotwireless get-service-profile \
    --endpoint-url https://api.iotwireless.region.amazonaws.com
    --id="1a2345ba-4c5d-67b0-ab67-e0c8342f2857"
```

A seguir, mostra um exemplo para o comando list-device-profiles.

```
aws iotwireless list-device-profiles \
    --endpoint-url https://api.iotwireless.region.amazonaws.com
```

IntegraçãoAWS IoT Core for LoRaWANEndpoints de API do plano de dados

AWS IoT Core for LoRaWANos pontos de extremidade do plano de dados consistem nos seguintes endpoints. Você obtém esses endpoints ao adicionar seu gateway aoAWS IoT Core for LoRaWAN. Para obter mais informações, consulte Adicionar um gateway aoAWS IoT Core for LoRaWAN (p. 1135).

- Pontos de extremidade do LoraWAN Network Server (LNS)

  Os endpoints do LNS são do formato`account-specific-prefix.lns.lorawan.region.amazonaws.com`. Você pode usar esse endpoint para estabelecer uma conexão para trocar mensagens de uplink e downlink LoRa.
- Pontos de extremidade do Servidor de Configuração e Atualização (CUPS)

  Os endpoints do CUPS são do formato`account-specific-prefix.cups.lorawan.region.amazonaws.com`. Você pode usar esse endpoint para gerenciamento de credenciais, configuração remota e atualização de firmware de gateways.

Para obter mais informações, consulte Usando protocolos CUPS e LNS (p. 1160).

Para encontrar os endpoints da API Data Plane para seuConta da AWSe Region, use oget-service-endpointComando da CLI mostrado aqui, ou o`GetServiceEndpoint`API REST. Para obter mais informações, consulteAWS IoT Core for LoRaWANEndpoints de API do plano de dados.

Você pode conectar seu gateway LoRaWAN no local para se comunicar comAWS IoT Core for LoRaWANEndpoints do . Para estabelecer essa conexão, primeiro conecte seu gateway local ao seuConta da AWSem sua VPC usando uma conexão VPN. Em seguida, você pode se comunicar com os pontos de extremidade da interface do plano de dados naAWS IoT Core for LoRaWANVPC que são alimentadas por privatelink.

Veja a seguir como integrar esses endpoints.

- Criar endpoint de interface VPC e zona hospedada privada (p. 1154)
- Use VPN para conectar gateways LoRa ao seuConta da AWS (p. 1157)

Criar endpoint de interface VPC e zona hospedada privada

AWS IoT Core for LoRaWANtem dois pontos de extremidade do plano de dados, ponto de extremidade do Servidor de Configuração e Atualização (CUPS) e o endpoint LoRaWAN Network Server (LNS). O processo de configuração para estabelecer uma conexão privatelink para ambos os endpoints é o mesmo, para que possamos usar o endpoint LNS para fins de ilustração.

Para seus endpoints de plano de dados, os gateways LoRa primeiro se conectam ao seuConta da AWSem sua Amazon VPC, que se conecta ao endpoint da VPC naAWS IoT Core for LoRaWANVPC.

Ao se conectar aos endpoints, os nomes DNS podem ser resolvidos em uma VPC, mas não podem ser resolvidos em várias VPCs. Para desativar o DNS privado ao criar o endpoint, desative oHabilitar o nome DNS. Você pode usar a zona hospedada privada para fornecer informações sobre como deseja que o Route 53 responda a consultas de DNS para suas VPCs. Para compartilhar sua VPC com um ambiente local, você pode usar umResolver Route 53 to facilita o DNS híbrido.

Para concluir esse procedimento, execute as seguintes etapas.

- Criar uma Amazon VPC e uma sub-rede (p. 1155)
- Criar um endpoint de interface da Amazon VPC (p. 1155)
- Configurar zona hospedada privada (p. 1155)
- Configurar resolvedor de entrada Route 53 (p. 1156)
- Próximas etapas (p. 1157)

Criar uma Amazon VPC e uma sub-rede

Você pode reutilizar a Amazon VPC e a sub-rede que você criou ao integrar o endpoint do plano de controle. Para obter mais informações, consulte Crie sua Amazon VPC e sub-rede (p. 1151).

Criar um endpoint de interface da Amazon VPC

Você pode criar um endpoint de VPC para sua VPC, que é semelhante a como você criaria um para o endpoint do plano de controle.

1. Navegue até o .VPC Endpoints doConsole e escolhaCriar endpoint.
2. NoCriar endpoint, especifique as seguintes informações.
   - SelecioneAWS service (Serviço da AWS)speloCategoria de serviço.
   - para oNome do serviço, pesquise digitando a palavra-chave**`lns`**. Na lista de`lns`serviços exibidos, escolha o endpoint da API do plano de dados LNS para sua região. O endpoint será do formato`com.amazonaws.`*`region`*`.lorawan.lns`.

     Note

     Se você estiver seguindo esse procedimento para o endpoint do CUPS, procure por`cups`. O endpoint será do formato`com.amazonaws.`*`region`*`.lorawan.cups`.
   - para oVPCeSub-redes, escolha a VPC na qual você deseja criar o endpoint e as Zonas de disponibilidade (AZs) nas quais você deseja criar a rede de endpoint.

     Note

     O`iotwireless`O serviço pode não oferecer suporte a todas as zonas de disponibilidade.
   - para oHabilitar o nome DNS, verifique seAtivar para este endpointnão está selecionado.

     Ao não selecionar essa opção, você pode desativar o DNS privado para o endpoint da VPC e usar a zona hospedada privada.
   - para oGrupo de segurança, escolha os grupos de segurança que você deseja associar às interfaces de rede do endpoint.
   - Opcionalmente, você pode adicionar ou remover tags. Tags são pares nome/valor usados para associar ao endpoint.
3. Para criar seu endpoint de VPC, escolhaCriar endpoint.

Configurar zona hospedada privada

Depois de criar o endpoint privatelink, noDetalhesNa guia do endpoint, você verá uma lista de nomes DNS. Você pode usar um desses nomes DNS para configurar sua zona hospedada privada. O nome DNS será do formato`vpce-`*`xxxx`*`.lns.lorawan.`*`region`*`.vpce.amazonaws.com`.

Criar a zona hospedada privada

Para criar a zona hospedada privada:

1. Navegue até o .Route 53 Zonas hospedadasConsole e escolhaCriar zona hospedada.
2. NoCriar zona hospedada, especifique as seguintes informações.

- para oNome de domínio, insira o nome completo do serviço para seu endpoint LNS,**`lns.lorawan.region.amazonaws.com`**.

  Note

  Se você estiver seguindo este procedimento para o endpoint do CUPS, insira**`cups.lorawan.region.amazonaws.com`**.
- para oTipo, escolhaZonas hospedadas privadas.
- Opcionalmente, você pode adicionar ou remover tags para associar à sua zona hospedada.

3. Para criar sua zona hospedada privada, escolhaCriar zona hospedada.

Para obter mais informações, consulteCriar uma zona hospedada privada.

Depois de criar uma zona hospedada privada, você pode criar um registro que informa ao DNS como você deseja que o tráfego seja roteado para esse domínio.

Criar um registro

Depois de criar uma zona hospedada privada, você pode criar um registro que informa ao DNS como você deseja que o tráfego seja roteado para esse domínio. Para criar um registro:

1. Na lista de zonas hospedadas exibidas, escolha a zona hospedada privada criada anteriormente e escolhaCriar registro.
2. Use o método do assistente para criar o registro. Se o console apresentar a você oCriação rápida, escolhaAlternar para assistente.
3. SelecioneRoteamento simplespeloPolítica de roteamentoe, depois, escolhaPróximo.
4. NoConfigurar registros, escolhaDefinir registro simples.
5. NoDefinir registro simples:
   - para oNome de registro, insira o alias doConta da AWS. Você obtém esse valor ao integrar seu gateway ou ao usar o`GetServiceEndpoint`API REST.
   - para oTipo de registro, mantenha o valor como`A - Routes traffic to an IPv4 address and some AWS resources`.
   - para oValor/Encaminhar tráfego para, escolhaAlias para VPC endpoint. Depois, escolha aRegiãoEm seguida, escolha o endpoint criado anteriormente, conforme descrito emCriar um endpoint de interface da Amazon VPC (p. 1155)na lista de endpoints exibidos.
6. SelecioneDefinir registro simplespara criar seu registro.

Configurar resolvedor de entrada Route 53

Para compartilhar um endpoint de VPC com um ambiente local, um Resolvedor Route 53 pode ser usado para facilitar o DNS híbrido. O resolvedor de entrada permitirá que você roteie o tráfego da rede local para os pontos de extremidade do plano de dados sem passar pela Internet pública. Para retornar os valores de endereço IP privado do seu serviço, crie o Resolvedor Route 53 na mesma VPC que o endpoint da VPC.

Ao criar o resolvedor de entrada, você só precisa especificar sua VPC e as sub-redes que você criou anteriormente em suas Zonas de disponibilidade (AZs). O Resolvedor Route 53 usa essas informações para atribuir automaticamente um endereço IP para rotear o tráfego para cada uma das sub-redes.

Para criar o resolvedor de entrada:

1. Navegue até o .Route 53 Endpoints de entradaConsole e escolhaCriar endpoint de entrada.

   Note

   Verifique se você está usando o mesmoRegião da AWSque você usou ao criar o endpoint e a zona hospedada privada.

2. NoCriar endpoint de entrada, especifique as seguintes informações.
   - Insira um nome paraNome do endpoint(por exemplo,**`VPC_A_Test`**).
   - para oVPC na região, escolha a mesma VPC usada ao criar o endpoint da VPC.
   - Configurar aGrupo de segurança para este endpointpara permitir o tráfego de entrada da rede local.
   - Para endereço IP, escolhaUse um endereço IP selecionado automaticamente.
3. SelecioneEnviarpara criar seu resolvedor de entrada.

Para este eample, vamos supor que os endereços IP`10.100.0.145`e`10.100.192.10`foram atribuídos para o Resolvedor Route 53 de entrada para tráfego de roteamento.

Próximas etapas

Você criou a zona hospedada privada e um resolvedor de entrada para rotear o tráfego para suas entradas DNS. Agora é possível usar uma VPN do Site-to-Site VPN ou um endpoint do Client VPN. Para obter mais informações, consulte Use VPN para conectar gateways LoRa ao seuConta da AWS (p. 1157).

Use VPN para conectar gateways LoRa ao seuConta da AWS

Para conectar seus gateways no local ao seuConta da AWSVocê pode usar uma conexão do Site-to-Site VPN ou um endpoint do Client VPN.

Antes de conectar seus gateways locais, você deve ter criado o endpoint da VPC e configurado uma zona hospedada privada e um resolvedor de entrada para que o tráfego dos gateways não passe pela Internet pública. Para obter mais informações, consulte Criar endpoint de interface VPC e zona hospedada privada (p. 1154).

Endpoint Site-to-Site VPN

Se você não tiver o hardware do gateway ou quiser testar a conexão VPN usando outraConta da AWS, é possível usar uma conexão do Site-to-Site VPN. Você pode usar o Site-to-Site VPN para se conectar aos endpoints da VPC do mesmoConta da AWSou outroConta da AWSque você pode estar usando em um diferenteRegião da AWS.

> **Note**
>
> Se você tiver o hardware de gateway com você e quiser configurar uma conexão VPN, recomendamos que você use a Client VPN. Para obter instruções, consulte Endpoint do cliente VPN (p. 1158).

Para configurar uma Site-to-Site VPN:

1. Crie outra VPC no site a partir do qual você deseja configurar a conexão. para o`VPC-A`, é possível reutilizar a VPC criada anteriormente. Para criar outra VPC (por exemplo,`VPC-B`), use um bloco CIDR que não se sobreponha ao bloco CIDR da VPC criada anteriormente.

   Para obter informações sobre a configuração das VPCs, siga as instruções descritas emAWSConfigurar a conexão do Site-to-Site VPN.

   > **Note**
   >
   > O método Site-to-Site VPN descrito no documento usa o OpenSwan para a conexão VPN, que suporta apenas um túnel VPN. Se você usar um software comercial diferente para a VPN, poderá configurar dois túneis entre os sites.
2. Depois de configurar a conexão VPN, atualize a`/etc/resolv.conf`ao adicionar o endereço IP do resolvedor de entrada do seuConta da AWS. Use esse endereço IP para o nameserver. Para obter informações sobre como obter esse endereço IP, consulteConfigurar resolvedor de entrada Route 53 (p. 1156). Para este exemplo, podemos usar o endereço IP`10.100.0.145`que foi atribuído quando você criou o Resolvedor Route 53.

```
options timeout:2 attempts:5
; generated by /usr/sbin/dhclient-script
search region.compute.internal
nameserver 10.100.0.145
```

3. Agora podemos testar se a conexão VPN usa aAWS PrivateLinkendpoint em vez de passar pela internet pública usando um`nslookup`comando. Veja a seguir um exemplo de execução do comando.

```
nslookup account-specific-prefix.lns.lorawan.region.amazonaws.com
```

A seguir, mostra um exemplo de saída de execução do comando, que mostra um endereço IP privado indicando que a conexão foi estabelecida com oAWS PrivateLinkEndpoint LNS.

```
Server: 10.100.0.145
Address: 10.100.0.145

Non-authoritative answer:
Name: https://xxxxx.lns.lorawan.region.amazonaws.com
Address: 10.100.0.204
```

Para obter informações sobre como usar uma conexão do Site-to-Site VPN, consulteComo funciona o Site-to-Site VPN.

Endpoint do cliente VPN

O AWS Client VPN é um serviço de VPN gerenciado no cliente que protege o acesso aos recursos da AWS e da sua rede local. A seguir, mostra a arquitetura do serviço VPN do cliente.

Para estabelecer uma conexão VPN com um endpoint do Client VPN:

1. Crie um endpoint do Client VPN seguindo as instruções descritas emConceitos básicos deAWS Client VPN.
2. Faça login na rede local (por exemplo, um roteador Wi-Fi) usando a URL de acesso desse roteador (por exemplo,`192.168.1.1`), e encontreO nome da raiz e senha.
3. Configure seu gateway LoraWAN seguindo as instruções na documentação do gateway e, em seguida, adicione seu gateway aoAWS IoT Core for LoRaWAN. Para obter informações sobre como adicionar seu gateway, consulteIncorporar seus gateways paraAWS IoT Core for LoRaWAN (p. 1133).
4. Verifique se o firmware do gateway está atualizado. Se o firmware estiver desatualizado, você poderá seguir as instruções fornecidas na rede local para atualizar o firmware do gateway. Para obter mais

informações, consulte Atualize o firmware do gateway usando o serviço CUPS comAWS IoT Core for LoRaWAN (p. 1162).

5. Verifique se o OpenVPN foi habilitado. Se ele tiver sido ativado, vá para a próxima etapa para configurar o cliente OpenVPN dentro da rede local. Se não tiver sido habilitado, siga as instruções emGuia para instalar o OpenVPN para OpenWrt.

   Note

   Neste exemplo, usamos o OpenVPN. Você pode usar outros clientes VPN, comoAWS VPNouAWS Direct ConnectPara configurar a conexão Client VPN.

6. Configure o cliente OpenVPN com base nas informações da configuração do cliente e como você pode usarCliente OpenVPN usando LuCi.
7. SSH para a rede local e atualize `o/etc/resolv.conf`ao adicionar o endereço IP do resolvedor de entrada em seuConta da AWS(`10.100.0.145`).
8. Para o tráfego do gateway usarAWS PrivateLinkpara se conectar ao endpoint, substitua a primeira entrada DNS do gateway pelo endereço IP do resolvedor de entrada.

Para obter informações sobre como usar uma conexão do Site-to-Site VPN, consulteConceitos básicos do Client VPN.

Connect a endpoints LNS e CUPS VPC

A seguir, mostra como você pode testar sua conexão com os endpoints LNS e CUPS VPC.

Testar o endpoint CUPS

Para testar oAWS PrivateLinkPara conexão com o endpoint do CUPS do gateway LoRa, execute o seguinte comando:

```
curl -k -v -X POST https://xxxx.cups.region.iotwireless.iot:443/update-info
     --cacert cups.trust --cert cups.crt --key cups.key --header "Content-Type:
 application/json"
     --data '{
               "router": "xxxxxxxxxxxxx",
               "cupsUri": "https://xxxx.cups.lorawan.region.amazonaws.com:443",
               "cupsCredCrc":1234, "tcCredCrc":552384314
              }'
      —output cups.out
```

Testar endpoint do LNS

Para testar seu endpoint LNS, primeiro provisione um dispositivo LoRaWAN que funcionará com seu gateway sem fio. Em seguida, você pode adicionar seu dispositivo e executar ounirapós o qual você pode começar a enviar mensagens de uplink.

Como gerenciar gateways comAWS IoT Corepara LoRaWAN

Os gateways atuam como uma ponte e transportam dados do dispositivo LoRaWAN de e para um Servidor de Rede, geralmente em redes de alta largura de banda, como Wi-Fi, Ethernet ou Celular. Gateways LoRaWAN conectam dispositivos sem fio aAWS IoT Core for LoRaWAN.

A seguir estão algumas considerações importantes ao usar seus gateways comAWS IoT Core for LoRaWAN. Para obter informações sobre como adicionar seu gateway aoAWS IoT Core for LoRaWAN, consulteIncorporar seus gateways paraAWS IoT Core for LoRaWAN (p. 1133).

Requisito de software LoRa Basics Station

Para conectar o aoAWS IoT Core for LoRaWAN, seu gateway LorAwan deve ter um software chamadoEstação Básica LoRacorrendo nele. LoRa Basics Station é um software de código aberto mantido pela Semtech Corporation e distribuído por seusGitHubrepositório.AWS IoT Core for LoRaWANSuporta LoRa Basics Station versão 2.0.4 e posterior.

Usando gateways qualificados a partir doAWSCatálogo de dispositivos para parceiros

OAWSCatálogo de dispositivos para parceiroscontém gateways e kits de desenvolvedor qualificados para uso comAWS IoT Core for LoRaWAN. Recomendamos que você use esses gateways qualificados porque você não precisa modificar o software de incorporação para conectar os gateways aoAWS IoT Core. Esses gateways já têm uma versão do software BasicStation compatível comAWS IoT Core for LoRaWAN.

> Note
>
> Se você tiver um gateway que não esteja listado no Catálogo de parceiros como um gateway qualificado comAWS IoT Core for LoRaWAN, você ainda poderá usá-lo se o gateway estiver executando o software LoRa Basics Station com a versão 2.0.4 e posterior. Certifique-se de usarAutenticação de cliente e servidor TLSpara autenticar seu gateway LoraWAN.

Usando protocolos CUPS e LNS

O software LoRa Basics Station contém dois subprotocolos para conectar gateways a servidores de rede, protocolos LoraWAN Network Server (LNS) e Configuration and Update Server (CUPS).

O protocolo LNS estabelece uma conexão de dados entre um gateway compatível com LoRa Basics Station e um servidor de rede. As mensagens de uplink e downlink LoRa são trocadas por meio dessa conexão de dados por WebSockets seguros.

O protocolo CUPS permite o gerenciamento de credenciais e a configuração remota e a atualização de firmware dos gateways.AWS IoT Core for LoRaWANfornece endpoints LNS e CUPS para ingestão de dados LoRaWAN e gerenciamento de gateway remoto, respectivamente.

Para obter mais informações, consulteProtocolo LNSeProtocolo CUPS.

Configure as subbandas e os recursos de filtragem do gateway

Os gateways LorAwan executam umEstação Básica LoRasoftware que permite que os gateways se conectem aAWS IoT Core for LoRaWAN. Para conectar o aoAWS IoT Core for LoRaWAN, seu gateway LoRa primeiro consulta o servidor CUPS para o endpoint LNS e, em seguida, estabelece uma conexão de dados WebSockets com esse endpoint. Depois que a conexão for estabelecida, os quadros de uplink e downlink podem ser trocados por essa conexão.

Filtragem de quadros de dados LoRa recebidos pelo gateway

Depois que seu gateway LoraWAN estabelecer uma conexão com o endpoint,AWS IoT Core for LoRaWANresponde com um`router_config`mensagem que especifica um conjunto de parâmetros para a configuração do gateway LoRa, incluindo parâmetros de filtragem`NetIDeJoinEui`. Para obter mais informações sobre`router_config`e como uma conexão é estabelecida com o LoraWAN Network Server (LNS), consulteProtocolo LNS.

```
{
"msgtype"    : "router_config"
```

```
"NetID"      : [ INT, .. ]
"JoinEui"    : [ [INT,INT], .. ] // ranges: beg,end inclusive
"region"     : STRING             // e.g. "EU863", "US902", ..
"hwspec"     : STRING
"freq_range" : [ INT, INT ]       // min, max (hz)
"DRs"        : [ [INT,INT,INT], .. ]   // sf,bw,dnonly
"sx1301_conf": [ SX1301CONF, .. ]
"nocca"      : BOOL
"nodc"       : BOOL
"nodwell"    : BOOL
}
```

Os gateways transportam dados do dispositivo LoRaWAN de e para o LNS geralmente em redes de alta largura de banda, como Wi-Fi, Ethernet ou Celular. Os gateways geralmente captam todas as mensagens e passam pelo tráfego que chega a eleAWS IoT Core for LoRaWAN. No entanto, você pode configurar os gateways para filtrar parte do tráfego de dados do dispositivo, o que ajuda a economizar o uso da largura de banda e reduz o fluxo de tráfego entre o gateway e o LNS.

Para configurar o gateway LoRa para filtrar os quadros de dados, você pode usar os parâmetros`NetID`e`JoinEui`no`router_config`mensagem.`NetID`é uma lista de valores NetID que são aceitos. Qualquer quadro de dados LoRa portando um quadro de dados diferente daqueles listados será descartado.`JoinEui`é uma lista de pares de valores inteiros que codificam intervalos de valores de JoineUI. Os quadros de solicitação de junção serão descartados pelo gateway, a menos que o campo`JoinEui`na mensagem está dentro do intervalo [BegEUI, EndeUI].

Canais de frequência e subbandas

Para regiões de RF US915 e AU915, os dispositivos sem fio têm opções de 64 canais de uplink de 125kHz e 8 500kHz para acessar as redes LoRaWAN usando os gateways LoRa. Os canais de frequência de uplink são divididos em 8 subbandas, cada uma com 8 canais de 125kHz e um canal de 500kHz. Para cada gateway regular na região AU915, uma ou mais subbandas terão suporte.

Alguns dispositivos sem fio não conseguem pular entre subbandas e usar os canais de frequência em apenas uma subbanda quando conectados aoAWS IoT Core for LoRaWAN. Para que os pacotes de uplink desses dispositivos sejam transmitidos, configure os gateways LoRa para usar essa subbanda específica. Para gateways em outras regiões de RF, como EU868, essa configuração não é necessária.

Configure seu gateway para usar filtragem e subbandas usando o console

Você pode configurar seu gateway para usar uma subbanda específica e também habilitar a capacidade de filtrar os quadros de dados LoRa. Para especificar esses parâmetros usando o console:

1. Navegue até o .AWS IoT Core for LoRaWAN Gateways doPágina doAWS IoTConsole e escolhaAdicionar gateway.
2. Especifique os detalhes do gateway, como oEui do Gateway,Banda de frequência (RFRegion)e um opcionalName (Nome)eDescriçãoe escolha se deseja associar umAWS IoTcoisa para o seu gateway. Para obter mais informações sobre como adicionar um gateway, consulteAdicionar um gateway usando o console (p. 1135).
3. NoConfiguração LoRaWAN, você pode especificar as subbandas e informações de filtragem.
   - `SubBands`: Para adicionar uma subbanda, escolhaAdicionar subbandae especifique uma lista de valores inteiros que indicam quais subbandas são suportadas pelo gateway. O`SubBands`Somente o parâmetro pode ser configurado no`RfRegion`US915 e AU915 e deve ter valores no intervalo`[1,8]`Em uma dessas regiões compatíveis.
   - `NetIdFilters`: Para filtrar quadros de uplink, escolhaAdicionar NetIDe especifique uma lista de valores de string que o gateway usa. O NetID do quadro de uplink de entrada do dispositivo sem fio deve corresponder a pelo menos um dos valores listados, caso contrário, o quadro será descartado.

- `JoinEuiFilters`: EscolhaAdicionar a gama JoineUIe especifique uma lista de pares de valores de string que um gateway usa para filtrar quadros LoRa. O valor JoineUI especificado como parte da solicitação de junção do dispositivo sem fio deve estar dentro do intervalo de pelo menos um dos valores JoineUirange, cada um listado como um par de [BegEUI, EndeUI], caso contrário, o quadro será descartado.

4. Em seguida, você pode continuar configurando seu gateway seguindo as instruções descritas emAdicionar um gateway usando o console (p. 1135).

Depois de adicionar um gateway, noAWS IoT Core for LoRaWAN Gateways doPágina doAWS IoTconsole, se você selecionar o gateway que você adicionou, poderá ver o`SubBands`e filtros`NetIdFilters`e`JoinEuiFilters`noDetalhes específicos do LoRaWANSeção da página de detalhes do Gateway.

Configure seu gateway para usar filtragem e subbandas usando a API

Você pode usar oCreateWirelessGatewayAPI que você usa para criar um gateway para configurar as subbandas que deseja usar e habilitar o recurso de filtragem. Usar o`CreateWirelessGateway`API, você pode especificar as subbandas e os filtros como parte das informações de configuração do gateway que você fornece usando o`LoRaWAN`campo. A tabela a seguir mostra o token de solicitação que inclui essas informações.

```
POST /wireless-gateways HTTP/1.1
Content-type: application/json

{
"Arn": "arn:aws:iotwireless:us-east-1:400232685877aa:WirelessGateway/
       a11e3d21-e44c-471c-afca-6716c228336a",
"Description": "Using my first LoRaWAN gateway",
   "LoRaWAN": {
      "GatewayEui": "a1b2c3d4567890ab",
      "JoinEuiFilters": [
        ["0000000000000001", "00000000000000ff"],
        ["000000000000ff00", "000000000000ffff"]
      ],
      "NetIdFilters": ["000000", "000001"],
      "RfRegion": "US915",
      "SubBands": [2]
   },
   "Name": "myFirstLoRaWANGateway"
   "ThingArn": null,
   "ThingName": null
}
```

Você também pode usar oUpdateWirelessGatewayAPI para atualizar os filtros, mas não as subbandas. Se o`JoinEuiFilters`e`NetIdfilters`os valores são nulos, significa que não há atualização para os campos. Se os valores não forem nulos e listas vazias forem incluídas, a atualização será aplicada. Para obter os valores dos campos especificados, use oGetWirelessGatewayAPI.

Atualize o firmware do gateway usando o serviço CUPS comAWS IoT Core for LoRaWAN

OEstação Básica LoRao software executado no gateway fornece gerenciamento de credenciais e interface de atualização de firmware usando o protocolo CUPS (Configuration and Update Server). O protocolo CUPS fornece entrega segura de atualização de firmware com assinaturas ECDSA.

Você terá que atualizar com frequência o firmware do gateway. Você pode usar o serviço CUPS com oAWS IoT Core for LoRaWANpara fornecer atualizações de firmware para o gateway onde as atualizações também podem ser assinadas. Para atualizar o firmware do gateway, você pode usar o SDK ou a CLI, mas não o console.

O processo de atualização leva cerca de 45 minutos para ser concluído. Pode demorar mais tempo se você estiver configurando seu gateway pela primeira vez para se conectar aoAWS IoT Core for LoRaWAN. Os fabricantes de gateway geralmente fornecem seus próprios arquivos de atualização de firmware e assinaturas para que você possa usá-lo e prosseguirFaça upload do arquivo de firmware em um bucket do S3 e adicione uma função do IAM (p. 1166).

Se você não tiver os arquivos de atualização de firmware, consulteGerar o arquivo de atualização de firmware e a assinatura (p. 1163)para um exemplo que você pode usar para se adaptar ao seu aplicativo.

Para executar a atualização de firmware do gateway:

- Gerar o arquivo de atualização de firmware e a assinatura (p. 1163)
- Faça upload do arquivo de firmware em um bucket do S3 e adicione uma função do IAM (p. 1166)
- Agende e execute a atualização do firmware usando uma definição de tarefa (p. 1169)

Gerar o arquivo de atualização de firmware e a assinatura

As etapas neste procedimento são opcionais e dependem do gateway que você está usando. Os fabricantes de gateway fornecem sua própria atualização de firmware na forma de um arquivo de atualização ou de um script e o Basics Station executa esse script em segundo plano. Nesse caso, você provavelmente encontrará o arquivo de atualização de firmware nas notas de versão do gateway que você está usando. Em vez disso, você pode usar esse arquivo de atualização ou script e prosseguir paraFaça upload do arquivo de firmware em um bucket do S3 e adicione uma função do IAM (p. 1166).

Se você não tiver esse script, a seguir mostra os comandos a serem executados para gerar o arquivo de atualização do firmware. As atualizações também podem ser assinadas para garantir que o código não tenha sido alterado ou corrompido e os dispositivos executem o código publicado apenas por autores confiáveis.

Neste procedimento, você irá:

- Gerar o arquivo de atualização de firmware (p. 1163)
- Gerar assinatura para a atualização de firmware (p. 1165)
- Revise os próximos passos (p. 1166)

Gerar o arquivo de atualização de firmware

O software LoRa Basics Station executado no gateway é capaz de receber atualizações de firmware na resposta do CUPS. Se você não tiver um script fornecido pelo fabricante, consulte o seguinte script de atualização de firmware escrito para o RakWireless Gateway baseado em Raspberry Pi. Temos um script base e o novo binário de estação, arquivo de versão `station.conf`estão anexados a ele.

> **Note**
>
> O script é específico para o Gateway RakWireless, então você terá que adaptá-lo ao aplicativo, dependendo do gateway que você estiver usando.

Script base

A seguir mostra um exemplo de script base para o RakWireless Gateway baseado em Raspberry Pi Você pode salvar os seguintes comandos em um arquivo`base.sh`e, em seguida, execute o script no terminal no navegador da Web do Raspberry Pi.

```
*#!/bin/bash*
execution_folder=/home/pi/Documents/basicstation/examples/aws_lorawan
station_path="$execution_folder/station"
version_path="$execution_folder/version.txt"
station_conf_path="$execution_folder/station_conf"

# Function to find the Basics Station binary at the end of this script
# and store it in the station path
function prepare_station()
{
 match=$(grep --text --line-number '^STATION:$' $0 | cut -d ':' -f 1)
 payload_start=$((match + 1))
 match_end=$(grep --text --line-number '^END_STATION:$' $0 | cut -d ':' -f 1)
 payload_end=$((match_end - 1))
 lines=$(($payload_end-$payload_start+1))
 head -n $payload_end $0 | tail -n $lines  > $station_path
}

# Function to find the version.txt at the end of this script
# and store it in the location for version.txt
function prepare_version()
{
  match=$(grep --text --line-number '^VERSION:$' $0 | cut -d ':' -f 1)
  payload_start=$((match + 1))
  match_end=$(grep --text --line-number '^END_VERSION:$' $0 | cut -d ':' -f 1)
  payload_end=$((match_end - 1))
  lines=$(($payload_end-$payload_start+1))
  head -n $payload_end $0 | tail -n $lines  > $version_path
}

# Function to find the version.txt at the end of this script
# and store it in the location for version.txt
function prepare_station_conf()
{
 match=$(grep --text --line-number '^CONF:$' $0 | cut -d ':' -f 1)
 payload_start=$((match + 1))
 match_end=$(grep --text --line-number '^END_CONF:$' $0 | cut -d ':' -f 1)
 payload_end=$((match_end - 1))
 lines=$(($payload_end-$payload_start+1))
 head -n $payload_end $0 | tail -n $lines  > $station_conf_path
}

# Stop the currently running Basics station so that it can be overwritten
# by the new one
killall station

# Store the different files
prepare_station
prepare_versionp
prepare_station_conf

# Provide execute permission for Basics station binary
chmod +x $station_path

# Remove update.bin so that it is not read again next time Basics station starts
rm -f /tmp/update.bin

# Exit so that rest of this script which has binaries attached does not get executed
exit 0
```

Adicionar script de carga

Ao script base, anexamos o binário Basics Station, o version.txt que identifica a versão para a qual atualizar, `estation.conf`em um script chamado`addpayload.sh`. Em seguida, execute esse script.

```
*#!/bin/bash
*
base.sh > fwstation

# Add station
echo "STATION:" >> fwstation
cat $1 >> fwstation
echo "" >> fwstation
echo "END_STATION:" >> fwstation

# Add version.txt
echo "VERSION:" >> fwstation
cat $2 >> fwstation
echo "" >> fwstation
echo "END_VERSION:" >> fwstation

# Add station.conf
echo "CONF:" >> fwstation
cat $3 >> fwstation
echo "END_CONF:" >> fwstation

# executable
chmod +x fwstation
```

Depois de executar esses scripts, você pode executar o seguinte comando no terminal para gerar o arquivo de atualização de firmware,`fwstation`.

```
$ ./addpayload.sh station version.txt station.conf
```

Gerar assinatura para a atualização de firmware

O software LoRa Basics Station fornece atualizações de firmware assinadas com assinaturas ECDSA. Para oferecer suporte a atualizações assinadas, você precisará de:

- Uma assinatura que deve ser gerada por uma chave privada ECDSA e menos de 128 bytes.
- A chave privada que é usada para a assinatura e deve ser armazenada no gateway com o nome do arquivo do formato`sig-%d.key`. Recomendamos usar o nome do arquivo`sig-0.key`.
- Um CRC de 32 bits sobre a chave privada.

A assinatura e o CRC serão passados para oAWS IoT Core for LoRaWANAPIs. Para gerar os arquivos anteriores, você pode usar o seguinte script`gen.sh`que é inspirado pelobasicstationExemplo no repositório do GitHub.

```
*#!/bin/bash

*function ecdsaKey() {
    # Key not password protected for simplicity
    openssl ecparam -name prime256v1 -genkey | openssl ec -out $1
}

# Generate ECDSA key
ecdsaKey sig-0.prime256v1.pem

# Generate public key
openssl ec -in sig-0.prime256v1.pem -pubout -out sig-0.prime256v1.pub

# Generate signature private key
openssl ec -in sig-0.prime256v1.pub -inform PEM -outform DER -pubin | tail -c 64 >
 sig-0.key
```

```
# Generate signature
openssl dgst -sha512 -sign sig-0.prime256v1.pem $1 > sig-0.signature

# Convert signature to base64
openssl enc -base64 -in sig-0.signature -out sig-0.signature.base64

# Print the crc
crc_res=$(crc32 sig-0.key)printf "The crc for the private key=%d\n" $((16#$crc_res))

# Remove the generated files which won't be needed later
rm -rf sig-0.prime256v1.pem sig-0.signature sig-0.prime256v1.pub
```

A chave privada gerada pelo script deve ser salva no gateway. O arquivo de chaves está em formato binário.

```
./gen_sig.sh fwstation
read EC key
writing EC key
read EC key
writing EC key
read EC key
writing EC key
The crc for the private key=3434210794

$ cat sig-0.signature.base64
MEQCIDPY/p2ssgXIPNCOgZr+NzeTLpX+WfBo5tYWbh5pQWN3AiBROen+XlIdMScv
AsfVfU/ZScJCalkVNZh4esyS8mNIgA==

$ ls sig-0.key
sig-0.key

$ scp sig-0.key pi@192.168.1.11:/home/pi/Documents/basicstation/examples/iotwireless
```

Revise os próximos passos

Agora que você gerou o firmware e a assinatura, vá para o próximo tópico para fazer upload do arquivo de firmware,`fwstation`, para um bucket do Amazon S3. O bucket é um contêiner que armazenará o arquivo de atualização de firmware como um objeto. Você pode adicionar uma função do IAM que dará ao servidor CUPS permissão para ler o arquivo de atualização de firmware no bucket do S3.

Faça upload do arquivo de firmware em um bucket do S3 e adicione uma função do IAM

Você pode usar o Amazon S3 para criar umbalde, que é um contêiner que pode armazenar seu arquivo de atualização de firmware. Você pode fazer upload do arquivo para o bucket do S3 e adicionar uma função do IAM que permita que o servidor CUPS leia o arquivo de atualização do bucket. Para obter mais informações sobre o Amazon S3, consulteConceitos básicos do Amazon S3.

O arquivo de atualização de firmware que você deseja carregar depende do gateway que você está usando. Se você seguiu um procedimento semelhante ao descrito emGerar o arquivo de atualização de firmware e a assinatura (p. 1163), você enviará o`fwstation`arquivo gerado pela execução dos scripts.

Este procedimento leva cerca de 20 minutos para ser concluído.

Para carregar o arquivo de firmware:

- Crie um bucket do Amazon S3 e carregue o arquivo de atualização (p. 1167)
- Crie uma função do IAM com permissões para ler o bucket do S3 (p. 1167)
- Revise os próximos passos (p. 1169)

Crie um bucket do Amazon S3 e carregue o arquivo de atualização

Você criará um bucket do Amazon S3 usando oAWS Management Consolee, em seguida, carregue seu arquivo de atualização de firmware no bucket.

Crie um bucket do S3.

Para criar um bucket do S3, abra oConsole do Amazon S3. Inicie a sessão, se ainda não o fez e execute as etapas a seguir:

1. Selecione Create bucket (Criar bucket).
2. Insira um nome exclusivo e significativo para oBucket name, (por exemplo:`iotwirelessfwupdate`). Para obter a convenção de nomenclatura recomendada para seu bucket, consultehttps://docs.aws.amazon.com/AmazonS3/latest/userguide/bucketnamingrules.html.
3. Certifique-se de que você selecionou oRegião da AWSselecionado como o que você usou para criar seu gateway e dispositivo LoraWAN, e oBlock all public access (Bloquear todo o acesso público)é selecionada para que seu bucket use as permissões padrão.
4. SelecioneHabilitar opeloBucket versioningO que ajudará você a manter várias versões do arquivo de atualização de firmware no mesmo bucket.
5. ConfirmarCriptografia do lado do servidoré definido comoDesabilitar oe escolhaCriar bucket.

Faça upload do arquivo de atualização de firmware

Agora você pode ver o bucket na lista de buckets exibidos noAWS Management Console. Escolha seu bucket e conclua as etapas a seguir para fazer upload do arquivo.

1. Escolha o bucket e, em seguida, escolhaCarregar.
2. SelecioneAdicionar arquivoe depois faça o upload do arquivo de atualização do firmware. Se você seguiu o procedimento descrito emGerar o arquivo de atualização de firmware e a assinatura (p. 1163), você enviará o`fwstation`, caso contrário, carregue o arquivo fornecido pelo fabricante do gateway.
3. Certifique-se de que todas as configurações estejam definidas como padrão. Certifique-se de queACLs predefinidas doé definido comoprivadoe escolhaCarregarpara fazer upload do arquivo.
4. Copie o URI do S3 do arquivo que você enviou. Escolha seu bucket e você verá o arquivo que você carregou exibido na lista deObjetos. Escolha o arquivo e, em seguida, escolhaCopiar URI S3. O URI será algo como:`s3://iotwirelessfwupdate/fwstation`se você nomeou seu bucket semelhante ao exemplo descrito anteriormente (`fwstation`). Você usará o URI do S3 ao criar a função do IAM.

Crie uma função do IAM com permissões para ler o bucket do S3

Agora você criará uma função e uma política do IAM que darão ao CUPS a permissão para ler o arquivo de atualização de firmware do bucket do S3.

Crie uma política do IAM para sua função

Para criar uma política do IAM para seuAWS IoT Core for LoRaWANfunção de destino, abra aHub de políticas do console do IAMDepois, execute as etapas a seguir:

1. SelecioneCriar política, e escolha oJSONGuia.
2. Exclua qualquer conteúdo do editor e cole este documento de política. A política fornece as permissões para acessar o`iotwirelessbucket` e o arquivo de atualização do firmware,`fwstation`, armazenado dentro de um objeto.

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

```
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": [
                "s3:ListBucketVersions",
                "s3:ListBucket",
                "s3:GetObject"
            ],
            "Resource": [
                "arn:aws:s3:::iotwirelessfwupdate/fwstation",
                "arn:aws:s3:::iotwirelessfwupdate"
            ]
        }
    ]
}
```

3. SelecionePolítica de revisão, e emName (Nome), insira um nome para essa política (por exemplo,`IoTWirelessFwUpdatePolicy`). Você precisará desse nome para usar no próximo procedimento.
4. Escolha Create policy (Criar política).

Criar uma função do IAM com a política anexada

Agora você criará uma função do IAM e anexe a política criada anteriormente para acessar o bucket do S3. Abrir oHub de funções do console do IAMe execute as etapas a seguir:

1. Escolha Create role (Criar função).
2. DentroSelecione o tipo de entidade confiável, escolhaOutroConta da AWS.
3. DentroID da conta, insira seuConta da AWSID e, em seguida, escolhaPróximo: Permissions
4. Na caixa de pesquisa, insira o nome da política do IAM criada no procedimento anterior. Verifique a política do IAM (por exemplo,`IoTWirelessFwUpdatePolicy`) você criou anteriormente nos resultados da pesquisa e escolhê-lo.
5. Selecione Next (Próximo): Tagse depois escolhaPróximo: Análise.
6. DentroNome da função, insira o nome dessa função (por exemplo,`IoTWirelessFwUpdateRole`) e, depois, escolhaCriar função.

Editar relação de confiança da função do IAM

Na mensagem de confirmação exibida após a execução da etapa anterior, escolha o nome da função criada para editá-la. Você editará a função para adicionar a relação de confiança a seguir.

1. NoSummary (Resumo)Seção da função criada, escolha a opçãoRelações de confiançae, depois, escolhaEditar relação de confiança.
2. DentroDocumento da política, altere o`Principal`Propriedade parecida com este exemplo.

```
"Principal": {
    "Service": "iotwireless.amazonaws.com"
},
```

Depois de alterar o`Principal`, o documento de política completo deve ser semelhante a este exemplo.

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

```
      "Principal": {
        "Service": "iotwireless.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {}
    }
  ]
}
```

3. Para salvar as alterações e sair, selecioneAtualizar política de confiança.
4. Obtenha o ARN para sua função. Escolha sua função do IAM e, na seção Resumo, você verá umARN de função, por exemplo,`arn:aws:iam::123456789012:role/IoTWirelessFwUpdateRole`. Copie issoARN de função.

Revise os próximos passos

Agora que você criou o bucket do S3 e uma função do IAM que permite que o servidor CUPS leia o bucket do S3, vá para o próximo tópico para agendar e executar a atualização do firmware. Manter oS3 URIeARN de funçãoque você copiou anteriormente para que você possa inseri-los para criar uma definição de tarefa que será executada para executar a atualização do firmware.

Agende e execute a atualização do firmware usando uma definição de tarefa

Você pode usar uma definição de tarefa para incluir detalhes sobre a atualização do firmware e definir a atualização.AWS IoT Core for LoRaWANfornece uma atualização de firmware com base nas informações dos três campos a seguir associados ao gateway.

- Estação

  A versão e o tempo de compilação do software Basics Station. Para identificar essas informações, você também pode gerá-las usando o software Basics Station que está sendo executado pelo gateway (por exemplo,`2.0.5(rpi/std) 2021-03-09 03:45:09`).
- PackageVersion

  A versão do firmware, especificada pelo arquivo`version.txt`no gateway. Embora essas informações possam não estar presentes no gateway, recomendamos que elas sejam uma forma de definir a versão do firmware (por exemplo,`1.0.0`).
- Modelo

  A plataforma ou modelo que está sendo usado pelo gateway (por exemplo, Linux).

Este procedimento leva 20 minutos para ser concluído.

Para concluir este procedimento:

- Obtenha a versão atual em execução em seu gateway (p. 1169)
- Crie uma definição de tarefa de gateway sem fio (p. 1170)
- Execute a tarefa de atualização de firmware e acompanhe o progresso (p. 1172)

Obtenha a versão atual em execução em seu gateway

Para determinar a elegibilidade do gateway para uma atualização de firmware, o servidor CUPS verifica todos os três campos,`Station`,`PackageVersion`, e`Model`, para uma correspondência quando o gateway os apresenta durante uma solicitação do CUPS. Quando você usa uma definição de tarefa, esses campos são armazenados como parte do`CurrentVersion`campo.

Você pode usar oAWS IoT Core for LoRaWANAPI ouAWS CLIPara obter o`CurrentVersion`para seu gateway. Os comandos a seguir mostram como obter essas informações usando a CLI.

1. Se você já provisionou um gateway, você pode obter informações sobre o gateway usando ogateway get-wireless-comando.

```
aws iotwireless get-wireless-gateway \
    --identifier 5a11b0a85a11b0a8 \
        --identifier-type GatewayEui
```

   A seguir, você verá uma saída de amostra para o comando.

```
{
    "Name": "Raspberry pi",
    "Id": "1352172b-0602-4b40-896f-54da9ed16b57",
    "Description": "Raspberry pi",
    "LoRaWAN": {
        "GatewayEui": "5a11b0a85a11b0a8",
        "RfRegion": "US915"
    },
    "Arn": "arn:aws:iotwireless:us-
east-1:231894231068:WirelessGateway/1352172b-0602-4b40-896f-54da9ed16b57"
}
```

2. Usando o ID do gateway sem fio relatado pelo`get-wireless-gateway`Comando, você pode usar oget-wireless-gateway-firmware-informationcomando para obter o`CurrentVersion`.

```
aws iotwireless get-wireless-gateway-firmware-information \
    --id "3039b406-5cc9-4307-925b-9948c63da25b"
```

   A seguir mostra um exemplo de saída para o comando, com informações de todos os três campos exibidos pelo`CurrentVersion`.

```
{
    "LoRaWAN": {
        "CurrentVersion": {
            "PackageVersion": "1.0.0",
            "Model": "rpi",
            "Station": "2.0.5(rpi/std) 2021-03-09 03:45:09"
        }
    }
}
```

Crie uma definição de tarefa de gateway sem fio

Quando você cria a definição de tarefa, recomendamos que você especifique a criação automática de tarefas usando o parâmetro AutoCreateTasks.`AutoCreateTasks`aplica-se a qualquer gateway que tenha uma correspondência para todos os três parâmetros mencionados anteriormente. Se esse parâmetro estiver desativado, os parâmetros devem ser atribuídos manualmente ao gateway.

Você pode criar a definição de tarefa de gateway sem fio usando oAWS IoT Core for LoRaWANAPI ouAWS CLI. Os comandos a seguir mostram como criar a definição de tarefa usando a CLI.

1. Crie um arquivo,`input.json`, que conterá as informações a serem passadas para o`CreateWirelessGatewayTaskDefinition`API. No`input.json`, forneça as seguintes informações que você obteve anteriormente:
   - UpdateDataSource

Forneça o link para o objeto que contém o arquivo de atualização de firmware que você carregou para o bucket do S3. (Por exemplo:`s3://iotwirelessfwupdate/fwstation`.

- AtualizadoDataRole

  Forneça o link para o ARN de função da função do IAM que você criou, que fornece permissões para ler o bucket do S3. (Por exemplo:`arn:aws:iam::123456789012:role/IoTWirelessFwUpdateRole`.

- SigkeyCRC e UpdateSignature

  Essas informações podem ser fornecidas pelo fabricante do gateway, mas se você seguiu o procedimento descrito emGerar o arquivo de atualização de firmware e a assinatura (p. 1163), você encontrará essas informações ao gerar a assinatura.

- CurrentVersion

  Forneça o`CurrentVersion`saída que você obteve anteriormente executando o`get-wireless-gateway-firmware-information` comando.

```
cat input.json
```

A seguir, você verá o conteúdo do`input.json`file.

```
{
    "AutoCreateTasks": true,
    "Name": "FirmwareUpdate",
    "Update":
    {
        "UpdateDataSource" : "s3://iotwirelessfwupdate/fwstation",
        "UpdateDataRole" : "arn:aws:iam::123456789012:role/IoTWirelessFwUpdateRole",
        "LoRaWAN" :
        {
            "SigKeyCrc": 3434210794,
            "UpdateSignature": "MEQCIDPY/p2ssgXIPNCOgZr+NzeTLpX+WfBo5tYWbh5pQWN3AiBROen
+XlIdMScvAsfVfU/ZScJCalkVNZh4esyS8mNIgA==",
            "CurrentVersion" :
            {
            "PackageVersion": "1.0.0",
            "Model": "rpi",
            "Station": "2.0.5(rpi/std) 2021-03-09 03:45:09"
            }
        }
    }
}
```

2. Pass o`input.json`arquivo para ocriar-wireless-gateway-task-definição-tarefaComando para criar a definição de tarefa.

```
aws iotwireless create-wireless-gateway-task-definition \
    --cli-input-json file://input.json
```

A seguir, você verá a saída do comando.

```
{
    "Id": "4ac46ff4-efc5-44fd-9def-e8517077bb12",
    "Arn": "arn:aws:iotwireless:us-
east-1:231894231068:WirelessGatewayTaskDefinition/4ac46ff4-efc5-44fd-9def-e8517077bb12"
}
```

Execute a tarefa de atualização de firmware e acompanhe o progresso

O gateway está pronto para receber a atualização do firmware e, uma vez ligado, ele se conecta ao servidor CUPS. Quando o servidor CUPS encontra uma correspondência na versão do gateway, ele agenda uma atualização de firmware.

Uma tarefa é uma definição de tarefa em andamento. Conforme você especificou a criação automática de tarefas definindo`AutoCreateTasks`para`True`, a tarefa de atualização de firmware é iniciada assim que um gateway correspondente for encontrado.

Você pode acompanhar o andamento da tarefa usando o`GetWirelessGatewayTask`API. Quando você executa oget-wireless-gateway-taskcomando pela primeira vez, ele mostrará o status da tarefa como`IN_PROGRESS`.

```
aws iotwireless get-wireless-gateway-task \
    --id 1352172b-0602-4b40-896f-54da9ed16b57
```

A seguir, você verá a saída do comando.

```
{
    "WirelessGatewayId": "1352172b-0602-4b40-896f-54da9ed16b57",
    "WirelessGatewayTaskDefinitionId": "ec11f9e7-b037-4fcc-aa60-a43b839f5de3",
    "LastUplinkReceivedAt": "2021-03-12T09:56:12.047Z",
    "TaskCreatedAt": "2021-03-12T09:56:12.047Z",
    "Status": "IN_PROGRESS"
}
```

Quando você executar o comando na próxima vez, se a atualização do firmware entrar em vigor, ele mostrará os campos atualizados,`Package`,`Version`, e`Model`E o status da tarefa muda para`COMPLETED`.

```
aws iotwireless get-wireless-gateway-task \
    --id 1352172b-0602-4b40-896f-54da9ed16b57
```

A seguir, você verá a saída do comando.

```
{
    "WirelessGatewayId": "1352172b-0602-4b40-896f-54da9ed16b57",
    "WirelessGatewayTaskDefinitionId": "ec11f9e7-b037-4fcc-aa60-a43b839f5de3",
    "LastUplinkReceivedAt": "2021-03-12T09:56:12.047Z",
    "TaskCreatedAt": "2021-03-12T09:56:12.047Z",
    "Status": "COMPLETED"
}
```

Neste exemplo, mostramos a atualização de firmware usando o gateway RakWireless baseado em Raspberry Pi. O script de atualização de firmware interrompe o BasicStation em execução para armazenar o atualizado`Package`,`Version`, e`Model`campos para que o BasicStation tenha que ser reiniciado.

```
2021-03-12 09:56:13.108 [CUP:INFO] CUPS provided update.bin
2021-03-12 09:56:13.108 [CUP:INFO] CUPS provided signature len=70 keycrc=37316C36
2021-03-12 09:56:13.148 [CUP:INFO] ECDSA key#0 -> VERIFIED
2021-03-12 09:56:13.148 [CUP:INFO] Running update.bin as background process
2021-03-12 09:56:13.149 [SYS:VERB] /tmp/update.bin: Forked, waiting...
2021-03-12 09:56:13.151 [SYS:INFO] Process /tmp/update.bin (pid=6873) completed
2021-03-12 09:56:13.152 [CUP:INFO] Interaction with CUPS done - next regular check in 10s
```

Se a atualização do firmware falhar, você verá um status de`FIRST_RETRY`do servidor CUPS e o gateway envia a mesma solicitação. Se o servidor CUPS não consegue se conectar ao gateway após um`SECOND_RETRY`, ele mostrará um status de`FAILED`.

Depois que a tarefa anterior foi`COMPLETED`ou`FAILED`, exclua a tarefa antiga usando odelete-wireless-gateway-taskcomando antes de iniciar um novo.

```
aws iotwireless delete-wireless-gateway-task \
    --id 1352172b-0602-4b40-896f-54da9ed16b57
```

Gerenciamento de dispositivos com o AWS IoT Core for LoRaWAN

Dispositivos LoRaWAN se comunicam comAWS IoT Core for LoRaWANatravés de gateways LoraWAN. Como adicionar dispositivos aoAWS IoT Core for LoRaWANvamosAWS IoTprocessar as mensagens recebidas dos dispositivos para uso porAWS IoTe outros serviços.

A seguir estão algumas considerações importantes ao usar seus dispositivos comAWS IoT Core for LoRaWAN. Para obter informações sobre como adicionar o dispositivo aoAWS IoT Core for LoRaWANConsulte, consulteIncorporar seus dispositivosAWS IoT Core for LoRaWAN (p. 1139).

Considerações sobre o dispositivo

Ao selecionar um dispositivo que você deseja usar para se comunicar com oAWS IoT Core for LoRaWAN, considere o seguinte.

- Sensores disponíveis
- Capacidade da bateria
- Consumo de energia
- Custos
- Tipo de antena e faixa de transmissão

Usando dispositivos com gateways qualificados paraAWS IoT Core for LoRaWAN

Os dispositivos que você usa podem ser emparelhados com gateways sem fio qualificados para uso comAWS IoT Core for LoRaWAN. Você pode encontrar esses gateways e kits de desenvolvedor noAWSCatálogo de dispositivos de parceiros. Também recomendamos que considere a proximidade desses dispositivos com seus gateways. Para obter mais informações, consulteUsando gateways qualificados a partir doAWSCatálogo de dispositivos para parceiros (p. 1160)

Versão LoRaWAN

AWS IoT Core for LoRaWANsuporta todos os dispositivos que estejam em conformidade com as especificações 1.0.x ou 1.1 LoRaWAN padronizadas por LoRa Aliança.

Modos de ativação

Antes que seu dispositivo LoraWAN possa enviar dados de uplink, você deve concluir um processo chamadoativaçãoouunir. Para ativar seu dispositivo, você pode usar OTAA (ativação por via aérea) ou ABP (Ativação por personalização). Recomendamos que você use o OTAA para ativar seu dispositivo porque novas chaves de sessão são geradas para cada ativação, o que o torna mais seguro.

A especificação do seu dispositivo sem fio é baseada na versão e no modo de ativação do LoRaWAN, que determina as chaves raiz e as chaves de sessão geradas para cada ativação. Para obter mais informações, consulteAdicione a especificação do dispositivo sem fio aoAWS IoT Core for LoRaWANUsar o console (p. 1140)

Classes de dispositivos

Os dispositivos LoraWAN podem enviar mensagens de uplink a qualquer momento. Ouvir mensagens downlink consome capacidade da bateria e reduz a duração da bateria. O protocolo LoRaWAN especifica três classes de dispositivos LoRaWAN.

- Os dispositivos de classe A dormem a maior parte do tempo e ouvem mensagens de downlink apenas por um curto período de tempo. Esses dispositivos são principalmente sensores alimentados por bateria com uma vida útil da bateria de até 10 anos.
- Os dispositivos de classe B podem receber mensagens em slots de downlink agendados. Esses dispositivos são principalmente atuadores alimentados por bateria.
- Os dispositivos de classe C nunca dormem e ouvem continuamente as mensagens recebidas e, portanto, não há muito atraso no recebimento das mensagens. Esses dispositivos são principalmente atuadores alimentados por mains-powered.

Para obter mais informações sobre essas considerações sobre dispositivos sem fio, consulte os recursos mencionados emSaiba mais sobre LoRaWAN (p. 1128).

Gerencie a comunicação entre seus dispositivos LoraWAN eAWS IoT

Depois de conectar seu dispositivo LoraWAN aoAWS IoT Core for LoRaWAN, seus dispositivos podem começar a enviar mensagens para a nuvem. Mensagens de uplink são mensagens enviadas do seu dispositivo e recebidas porAWS IoT Core for LoRaWAN. Seus dispositivos LoraWAN podem enviar mensagens de uplink a qualquer momento, que são então encaminhadas para outrosAWS service (Serviço da AWS)s e aplicativos hospedados em nuvem. Mensagens enviadas deAWS IoT Core for LoRaWANe outrosAWS service (Serviço da AWS)s e aplicativos para seus dispositivos são chamados de mensagens de downlink.

A seguir, mostra como você pode visualizar e gerenciar mensagens de uplink e downlink enviadas entre seus dispositivos e a nuvem. Você pode manter uma fila de mensagens de downlink e enviar essas mensagens para seus dispositivos na ordem em que foram adicionadas à fila.

Tópicos

- Exibir o formato das mensagens de uplink enviadas de dispositivos LoraWAN (p. 1174)
- Enfileirar mensagens de downlink para enviar para dispositivos LoraWAN (p. 1177)

Exibir o formato das mensagens de uplink enviadas de dispositivos LoraWAN

Depois de conectar seu dispositivo LoraWAN aoAWS IoT Core for LoRaWAN, você pode observar o formato da mensagem de uplink que você receberá do seu dispositivo sem fio.

Antes que você possa observar as mensagens de uplink

Você deve ter integrado seu dispositivo sem fio e conectado o dispositivo aoAWS IoTpara que ele possa transmitir e receber dados. Para obter informações sobre como integrar-se ao dispositivo aoAWS IoT Core for LoRaWANConsulte, consulteIncorporar seus dispositivosAWS IoT Core for LoRaWAN (p. 1139).

O que as mensagens de uplink contêm?

Os dispositivos LoRawan se conectam aAWS IoT Core for LoRaWANusando gateways LoraWAN. A mensagem de uplink que você recebe do dispositivo conterá as seguintes informações.

- Dados de carga útil que correspondem à mensagem de carga útil criptografada enviada do dispositivo sem fio.
- Metadados sem fio que incluem:
  - Informações do dispositivo, como DeveUI, a taxa de dados e o canal de frequência no qual o dispositivo está operando.
  - Parâmetros adicionais opcionais e as informações do gateway para gateways conectados ao dispositivo. Os parâmetros do gateway incluem o EUI do gateway, o SNR e o RSSi.

  Usando os metadados sem fio, você pode obter informações úteis sobre o dispositivo sem fio e os dados transmitidos entre o dispositivo eAWS IoT. Por exemplo, você pode usar a`AckedMessageId`parâmetro para verificar se a última mensagem de downlink confirmada foi recebida pelo dispositivo. Opcionalmente, se você optar por incluir as informações do gateway, poderá identificar se deseja alternar para um canal de gateway mais forte que esteja mais próximo do dispositivo.

Como observar as mensagens de uplink?

Depois de integrar seu dispositivo, você pode usar oCliente de teste MQTT danoTestePágina doAWS IoTConsole para se inscrever no tópico que você especificou ao criar seu destino. Você começará a ver as mensagens depois que o dispositivo estiver conectado e começar a enviar dados de carga útil.

Este diagrama identifica os elementos-chave em um sistema LoRawan conectado aAWS IoT Core for LoRaWAN, que mostra o plano de dados primário e como os dados fluem pelo sistema.

Quando o dispositivo sem fio começar a enviar dados de uplink,AWS IoT Core for LoRaWANenvolve as informações de metadados sem fio com a carga útil e, em seguida, as envia para oAWSaplicativos.

Exemplo de mensagem de Uplink

O exemplo a seguir mostra o formato da mensagem de uplink recebida do dispositivo.

> **Note**
>
> Se seus dispositivos enviarem uma mensagem de uplink sem um valor para`Fport`,AWS IoT Core for LoRaWANadicionará o valor 225 ao`Fport`na mensagem de uplink recebida.

```
{
    "WirelessDeviceId": "5b58245e-146c-4c30-9703-0ca942e3ff35",
    "PayloadData": "Cc48AAAAAAAAAAA=",
    "WirelessMetadata":
    {
        "LoRaWAN":
        {
            "ADR": false,
            "Bandwidth": 125,
            "ClassB": false,
            "CodeRate": "4/5",
            "DataRate": "0",
            "DevAddr": "00b96cd4",
            "DevEui": "58a0cb000202c99",
            "FOptLen": 2,
            "FCnt": 1,
            "Fport": 136,
            "Frequency": "868100000",
            "Gateways": [
             {
                    "GatewayEui": "80029cfffe5cf1cc",
                    "Snr": -29,
                    "Rssi": 9.75
             }
             ],
            "MIC": "7255cb07",
            "MType": "UnconfirmedDataUp",
            "Major": "LoRaWANR1",
            "Modulation": "LORA",
            "PolarizationInversion": false,
            "SpreadingFactor": 12,
            "Timestamp": "2021-05-03T03:24:29Z"

        }
    }
}
```

A tabela a seguir mostra uma descrição dos campos usados nos metadados de uplink:

Campos de mensagem de uplink LoraWAN

Parâmetro	Descrição	Type	Obrigatório
`WirelessDeviceID`	ID do dispositivo sem fio enviando os dados.	String	Sim
`PayloadData`	A mensagem binária recebida do dispositivo, codificada em base64.	String	Sim
`WirelessMetadata`	Metadados sobre o dispositivo LoRaWAN e a solicitação de mensagem. Isso inclui informações como identificadores de dispositivo, taxa de dados e código, o carimbo de data/hora da mensagem, se o ADR (taxa de dados adaptativa) está habilitado e os metadados do gateway.	Enumeração	Não

Excluir metadados de gateway de metadados de uplink

Se você quiser excluir as informações de metadados do gateway de seus metadados de uplink, desative oAddGwMetadataQuando você cria o perfil de serviço. Para obter informações sobre como desativar esse parâmetro, consulteAdicionar perfis de serviço (p. 1143).

Nesse caso, você não verá a`Gateways`nos metadados de uplink, conforme ilustrado no exemplo a seguir.

```
{
    "WirelessDeviceId": "0d9a439b-e77a-4573-a791-49d5c0f4db95",
    "PayloadData": "AAAAAAAA//8=",
    "WirelessMetadata": {
        "LoRaWAN": {
            "ClassB": false,
            "CodeRate": "4/5",
            "DataRate": "1",
            "DevAddr": "01920f27",
            "DevEui": "ffffff10000163b0",
            "FCnt": 1,
            "FPort": 5,
            "Timestamp": "2021-04-29T05:19:43.646Z"
    }
  }
}
```

Enfileirar mensagens de downlink para enviar para dispositivos LoraWAN

Aplicativos hospedados na nuvem e outrosAWS service (Serviço da AWS)s pode enviar mensagens downlink para seus dispositivos sem fio. Mensagens downlink são mensagens enviadas deAWS IoT Core for LoRaWANPara o dispositivo sem fio. Você pode agendar e enviar mensagens de downlink para cada dispositivo que você tenha incorporadoAWS IoT Core for LoRaWAN.

Se você tiver vários dispositivos para os quais deseja enviar uma mensagem de downlink, poderá usar um grupo multicast. Os dispositivos em um grupo multicast compartilham o mesmo endereço multicast, que é distribuído para um grupo inteiro de dispositivos destinatários. Para obter mais informações, consulteCrie grupos multicast para enviar uma carga de downlink para vários dispositivos (p. 1180)

Como funciona uma fila de mensagens de downlink

A classe de dispositivo do seu dispositivo LoRaWAN determina como as mensagens na fila são enviadas para o dispositivo. Dispositivos de classe A enviam uma mensagem de uplink paraAWS IoT Core for LoRaWANpara indicar que o dispositivo está disponível para receber mensagens de downlink. Os dispositivos de classe B podem receber mensagens em slots de downlink regulares. Os dispositivos de classe C podem receber mensagens de downlink a qualquer momento. Para obter mais informações sobre classes de dispositivos, consulte.Classes de dispositivos (p. 1174).

A seguir mostra como as mensagens são enfileiradas e enviadas para seus dispositivos de classe A.

1. AWS IoT Core for LoRaWANbuffers a mensagem de downlink que você adicionou à fila com a porta do quadro, dados de carga útil e os parâmetros do modo de confirmação que você especificou usando oAWS IoTConsole ou oAWS IoT WirelessAPI.
2. Seu dispositivo LoraWAN envia uma mensagem de uplink para indicar que está online e pode começar a receber mensagens de downlink.
3. Se você adicionou mais de uma mensagem de downlink à fila,AWS IoT Core for LoRaWANenvia a primeira mensagem de downlink na fila para o dispositivo com o sinalizador de confirmação (ACK) definido.
4. Seu dispositivo envia uma mensagem de uplink paraAWS IoT Core for LoRaWANimediatamente, ou dorme até a próxima mensagem de uplink e inclui o sinalizador ACK na mensagem.
5. QuandoAWS IoT Core for LoRaWANrecebe a mensagem de uplink com o sinalizador ACK, ele limpa a mensagem de downlink da fila, indicando que seu dispositivo recebeu com êxito a mensagem de downlink. Se o sinalizador ACK estiver ausente da mensagem de uplink após a verificação três vezes, a mensagem será descartada.

Execute operações de fila de downlink usando o console do

Você pode usar oAWS Management Consolepara enfileirar mensagens de downlink e limpar mensagens individuais, ou toda a fila, conforme necessário. Para dispositivos de classe A, depois que um uplink é recebido do dispositivo para indicar que ele está online, as mensagens em fila são enviadas para o dispositivo. Depois que a mensagem é enviada, ela é removida automaticamente da fila.

Fila mensagens de downlink

Para criar uma fila de mensagens de downlink

1. Vá para oHub de dispositivos doAWS IoTconsolee escolha o dispositivo para o qual você deseja enfileirar mensagens de downlink.
2. NoMensagens de downlinkSeção da página de detalhes do dispositivo, escolhaFila mensagens de downlink.
3. Especifique os seguintes parâmetros para configurar sua mensagem de downlink:
   - FPort: Escolha a porta do quadro para o dispositivo se comunicarAWS IoT Core for LoRaWAN.
   - Carga útil: Especifique a mensagem de carga útil que você deseja enviar para o dispositivo. O tamanho máximo de carga é de 242 bytes. Se a taxa de dados adaptativa (ADR) estiver ativada,AWS IoT Core for LoRaWANo usa para escolher a taxa de dados ideal para o tamanho da carga útil. Você pode otimizar ainda mais a taxa de dados conforme necessário.
   - Modo de reconhecimento: Confirme se o dispositivo recebeu a mensagem de downlink. Se uma mensagem exigir esse modo, você verá uma mensagem de uplink com o sinalizador ACK no fluxo de dados e a mensagem será removida da fila.
4. Para adicionar sua mensagem de downlink à fila, escolhaEnviar.

Sua mensagem de downlink agora foi adicionada à fila. Se você não vir sua mensagem ou receber um erro, solucionar o erro conforme descrito emSolucionar problemas de erros de fila de mensagens de downlink (p. 1179).

> **Note**
>
> Depois que sua mensagem de downlink for adicionada à fila, você não poderá mais editar os parâmetrosFPort,Carga útil, eModo de reconhecimento. Para enviar uma mensagem de downlink com valores diferentes para esses parâmetros, você pode excluir essa mensagem e enfileirar uma nova mensagem de downlink com os valores de parâmetro atualizados.

A fila lista as mensagens de downlink que você adicionou. Para ver a carga útil das mensagens de uplink e downlink que são trocadas entre seus dispositivos eAWS IoT Core for LoRaWAN, você pode usar o analisador de rede. Para obter mais informações, consulteMonitorando sua frota de recursos sem fio em tempo real usando o analisador de rede (p. 1201)

Lista de fila de mensagens de downlink

A mensagem de downlink que você criou é adicionada à fila. Cada mensagem de downlink subsequente é adicionada à fila após essa mensagem. Você pode ver uma lista de mensagens de downlink noMensagens de downlinkSeção da página de detalhes do dispositivo. Depois que um uplink é recebido, as mensagens são enviadas para o dispositivo. Depois que uma mensagem de downlink for recebida pelo seu dispositivo, ela será removida da fila. A próxima mensagem então se move para cima na fila a ser enviada para o dispositivo.

Exclua mensagens de downlink individuais ou limpe a fila inteira

Cada mensagem de downlink é removida da fila automaticamente após ser enviada para o dispositivo. Você também pode excluir mensagens individuais ou limpar toda a fila de downlink. Essas ações não podem ser desfeitas.

- Se você encontrar mensagens na fila que não deseja enviar, escolha as mensagens e escolhaExcluir.
- Se você não quiser enviar mensagens da fila para o dispositivo, você pode limpar a fila inteira escolhendoLimpar fila de downlink.

Execute operações de fila de downlink usando a API

Você pode usar oAWS IoT WirelessAPI para enfileirar mensagens de downlink e limpar mensagens individuais, ou toda a fila, conforme necessário.

Fila mensagens de downlink

Para criar uma fila de mensagens de downlink, use o`SendDataToWirelessDevice`Operação da API ou o`send-data-to-wireless-device`Comando da CLI.

```
aws iotwireless send-data-to-wireless-device \
    --id "11aa5eae-2f56-4b8e-a023-b28d98494e49" \
    --transmit-mode "1" \
    --payload-data "SGVsbG8gVG8gRGV2c2lt" \
    --wireless-metadata LoRaWAN={FPort=1}
```

A saída da execução deste comando gera um`MessageId`para a mensagem de downlink. Em alguns casos, mesmo que você receba a`MessageId`, os pacotes podem ser descartados. Para obter mais informações sobre como você pode resolver o erro, consulteSolucionar problemas de erros de fila de mensagens de downlink (p. 1179).

```
{
    MessageId: "6011dd36-0043d6eb-0072-0008"
}
```

Listar mensagens de downlink na fila

Para listar todas as mensagens de downlink na fila, use o`ListQueuedMessages`Operação da API ou o`list-queued-messages`Comando da CLI.

```
aws iotwireless list-queued-messages
```

Por padrão, no máximo 10 mensagens de downlink são exibidas ao executar este comando.

Remova mensagens de downlink individuais ou limpe a fila inteira

Para remover mensagens individuais da fila ou limpar toda a fila, use o`DeleteQueuedMessages`Operação da API ou o`delete-queued-messages`Comando da CLI.

- Para remover mensagens individuais, forneça o`messageID`para mensagens que você deseja remover para seu dispositivo sem fio, especificadas pelo`wirelessDeviceId`.
- Para limpar toda a fila de downlink, especifique`messageID`como`*`para seu dispositivo sem fio, especificado pelo`wirelessDeviceId`.

Solucionar problemas de erros de fila de mensagens de downlink

Aqui estão algumas coisas para verificar se você não está vendo os resultados esperados:

- Mensagens de downlink não são exibidas noAWS IoTconsole

  Se você não vir sua mensagem de downlink na fila depois de adicioná-la conforme descrito emExecute operações de fila de downlink usando o console do (p. 1178), pode ser porque seu dispositivo não

concluiu um processo chamadoativaçãoouJunção de procedimentos. Este procedimento é concluído quando o dispositivo estiver conectado comAWS IoT Core for LoRaWAN. Para obter mais informações, consulte Adicione a especificação do dispositivo sem fio aoAWS IoT Core for LoRaWANUsar o console (p. 1140) .

Depois de integrar seu dispositivo paraAWS IoT Core for LoRaWAN, você pode monitorar seu dispositivo para verificar se a junção e a ingressar foram bem-sucedidas usando o analisador de rede ou o Amazon CloudWatch. Para obter mais informações, consulteMonitorar e registrar em log paraAWS IoT WirelessUsar o Amazon CloudWatch (p. 1226)

- Pacotes de mensagens de downlink ausentes ao usar a API

  Quando você usa o`SendDataToWirelessDevice`Operação da API, a API retorna um exclusivo`MessageId`. No entanto, ele não pode confirmar se seu dispositivo LoraWAN recebeu a mensagem de downlink. Os pacotes de downlink podem ser descartados em casos como quando o dispositivo não concluiu o procedimento de junção. Para obter mais informações sobre como resolver esse erro, consulte a seção anterior.
- Erro de ARN ausente ao enviar mensagem de downlink

  Ao enviar uma mensagem de downlink para o dispositivo a partir da fila, você pode receber um erro do Amazon Resource Name (ARN) ausente. Esse erro pode ocorrer porque o destino não foi especificado corretamente para o dispositivo que está recebendo a mensagem de downlink. Para resolver esse erro, verifique os detalhes do destino do seu dispositivo.

Crie grupos multicast para enviar uma carga de downlink para vários dispositivos

Para enviar uma carga de downlink para vários dispositivos, crie um grupo multicast. Usando multicast, uma origem pode enviar dados para um único endereço multicast, que é distribuído para um grupo inteiro de dispositivos destinatários.

Os dispositivos em um grupo multicast compartilham o mesmo endereço multicast, chaves de sessão e contador de quadros. Usando as mesmas chaves de sessão, os dispositivos em um grupo multicast podem descriptografar a mensagem quando uma transmissão de downlink é iniciada. Um grupo multicast só suporta downlink. Ele não confirma se a carga de downlink foi recebida pelos dispositivos.

comAWS IoT Core for LoRaWANgrupos multicast, você pode:

- Filtre sua lista de dispositivos usando o perfil do dispositivo, RFRelion ou classe de dispositivo e, em seguida, adicione esses dispositivos a um grupo multicast.
- Programe e envie uma ou mais mensagens de carga útil de downlink para dispositivos em um grupo multicast, dentro de uma janela de distribuição de 48 horas.
- Faça com que os dispositivos mudem temporariamente para o modo Classe B ou classe C no início da sessão multicast para receber a mensagem de downlink.
- Monitore a configuração do grupo multicast e o estado de seus dispositivos, além de solucionar problemas.
- Use Atualizações de Firmware Over-The-Air (FUOTA) para implantar com segurança atualizações de firmware em dispositivos em um grupo multicast.

AWS IoT Core for LoRaWANo suporte do FUOTA e grupos multicast é baseado noAliança LoRaseguintes especificações:

- Especificação de configuração de multicast remoto LoRaWAN, TS005-2.0.0
- Especificação de transporte de bloco de dados fragmentados LoRaWAN, TS004-2.0.0

- Especificação de sincronização do relógio da camada de aplicativo LoRaWAN, TS003-2.0.0

  Note

  AWS IoT Core for LoRaWANexecuta automaticamente a sincronização do relógio para o dispositivo de acordo com o LoRa Especificação da aliança. Uso de função`AppTimeReq`, ele responde a hora do lado do servidor aos dispositivos que o solicitam usando ClockSync Sinalização.

A seguir, mostra como criar seu grupo multicast e agendar uma mensagem de downlink.

Tópicos

- Preparar dispositivos para configuração multicast e FUOTA (p. 1181)
- Crie grupos multicast e adicione dispositivos ao grupo (p. 1183)
- Monitore e solucione problemas de status de seu grupo multicast e dispositivos no grupo (p. 1187)
- Agende uma mensagem de downlink para enviar para dispositivos em seu grupo multicast (p. 1189)

Preparar dispositivos para configuração multicast e FUOTA

Quando você adiciona seu dispositivo sem fio aoAWS IoT Core for LoRaWAN, Você pode preparar seu dispositivo sem fio para configuração multicast e FUOTA usando o console ou a CLI. Se estiver executando essa configuração pela primeira vez, recomendamos usar o console. Para gerenciar seu grupo multicast e adicionar ou remover vários dispositivos do grupo, recomendamos usar a CLI para gerenciar um grande número de recursos.

GenAppKey e FPorts

Quando você adicionar seu dispositivo sem fio, antes de adicionar seus dispositivos a grupos multicast ou executar atualizações FUOTA, configure os seguintes parâmetros. Antes de configurar esses parâmetros, certifique-se de que seus dispositivos suportam FUOTA e multicast e a especificação do dispositivo sem fio seja`OTAA v1.1`ou`OTAAv1.0.x`.

- `GenAppKey`: Para dispositivos que suportam o LoRaWAN versão 1.0.x e para usar grupos multicast, o`GenAppKey`é a chave raiz específica do dispositivo da qual as chaves de sessão do seu grupo multicast são derivadas.

  Note

  Para dispositivos LoRaWAN que usam a especificação sem fio`OTAA v1.1`, o`AppKey`é usado para o mesmo propósito que o`GenAppKey`.

  Para configurar os parâmetros para iniciar a transferência de dados,AWS IoT Core for LoRaWANdistribui chaves de sessão com os dispositivos finais. Para obter mais informações sobre as versões do LoraWAN, consulte.Versão LoRaWAN (p. 1173).

  Note

  AWS IoT Core for LoRaWANarmazena o`GenAppKey`As informações que você fornece em um formato criptografado.
- `FPorts`: De acordo com as especificações LOraWAN para grupos FUOTA e multicast,AWS IoT Core for LoRaWANAtribui os valores padrão para os seguintes campos da`FPorts`parâmetro . Se você já tiver atribuído qualquer um dos seguintes`FPort`valores, então você pode escolher um valor diferente que esteja disponível, de 1 a 223.
  - `Multicast`: 200

    Esse`FPort`value é usado para grupos multicast.

- `FUOTA`: 201

  Esse`FPort`valor é usado para FUOTA.
- `ClockSync`: 202

  Esse`FPort`valor é usado para a sincronização do relógio.

Perfis de dispositivos para multicast e FUOTA

No início de uma sessão multicast, uma janela de distribuição de classe B ou classe C é usada para enviar a mensagem de downlink para os dispositivos do seu grupo. Os dispositivos que você adiciona para multicast e FUOTA devem suportar os modos de operação classe B ou classe C. Dependendo da classe de dispositivo compatível com o dispositivo, escolha um perfil de dispositivo para o dispositivo que tenha os modos classe B ou classe C.

Para obter informações sobre perfis de dispositivos, consulteAdicionar perfis aoAWS IoT Core for LoRaWAN (p. 1141).

Prepare dispositivos para multicast e FUOTA usando o console

Para especificar os FPorts e GenAppKey parâmetros para configuração multicast e FUOTA usando o console:

1. Navegue até o .Hub de dispositivos doAWS IoTconsolee escolhaAdicionar dispositivo sem fio.
2. Selecione oEspecificação do dispositivo sem fio. Seu dispositivo deve usar OTAA para ativação do dispositivo. Quando você escolhe OTAA v1.0.x ou OTAA v1.1, umConfiguração FUOTA - opcionalA seção é exibida.
3. Insira os parâmetros EUI (Extended Unique Identifier) para o seu dispositivo sem fio.
4. Expandir oConfiguração FUOTA - opcionalseção e, em seguida, escolhaEste dispositivo suporta atualizações de firmware pelo ar (FUOTA). Agora, você pode inserir oFPortvalores para multicast, FUOTA e sincronização de clock. Se você escolheu`OTAA v1.0.x`para a especificação do dispositivo sem fio, insira oGenAppKey.
5. Adicionar o dispositivo aoAWS IoT Core for LoRaWANescolhendo seus perfis e um destino para rotear mensagens. Para o perfil do dispositivo vinculado ao dispositivo, certifique-se de selecionar um ou ambosSuporta classe BouSuporta Classe CModos.

> **Note**
>
> Para especificar os parâmetros de configuração do FUOTA, você deve usar oHub de dispositivos doAWS IoTconsole. Esses parâmetros não aparecerão se você integrar seus dispositivos usando oIntroPágina doAWS IoTconsole do .

Para obter mais informações sobre a especificação do dispositivo sem fio e a integração do dispositivo, consulteAdicione seu dispositivo sem fio aoAWS IoT Core for LoRaWAN (p. 1140).

> **Note**
>
> Você pode especificar esses parâmetros somente ao criar o dispositivo sem fio. Você não pode alterar ou especificar parâmetros ao atualizar um dispositivo existente.

Prepare dispositivos para multicast e FUOTA usando a operação da API

Para usar grupos multicast ou para executar atualizações FUOTA, configure esses parâmetros usando o`CreateWirelessDevice`Operação da API ou o`create-wireless-device`Comando da CLI. Além de especificar a chave do aplicativo e os parâmetros FPorts, certifique-se de que o perfil do dispositivo vinculado ao dispositivo suporta um ou ambos os modos de classe B ou classe C.

Você pode fornecer um`input.json`arquivo como entrada para o`create-wireless-device`comando.

```
aws iotwireless create-wireless-device \
    --cli-input-json file://input.json
```

em que:

Conteúdo da input.json

```
{
    "Description": "My LoRaWAN wireless device"
    "DestinationName": "IoTWirelessDestination"
    "LoRaWAN": {
        "DeviceProfileId": "ab0c23d3-b001-45ef-6a01-2bc3de4f5333",
        "ServiceProfileId": "fe98dc76-cd12-001e-2d34-5550432da100",
        "FPorts": {
            "ClockSync": 202,
            "Fuota": 201,
            "Multicast": 200
      },
        "OtaaV1_0_x": {
            "AppKey": "3f4ca100e2fc675ea123f4eb12c4a012",
            "AppEui": "b4c231a359bc2e3d",
            "GenAppKey": "01c3f004a2d6efffe32c4eda14bcd2b4"
        },
        "DevEui": "ac12efc654d23fc2"
    },
    "Name": "SampleIoTWirelessThing"
    "Type": LoRaWAN
}
```

Para obter informações sobre os comandos da CLI que você pode usar, consulte o, consulteAWS CLIreferência.

> **Note**
>
> Depois de especificar os valores desses parâmetros, você não poderá atualizá-los usando o`UpdateWirelessDevice`Operação da API. Em vez disso, é possível criar um novodispositivoCom os valores dos parâmetros`GenAppKey`e`FPorts`.

Para obter informações sobre os valores especificados para esses parâmetros, você pode usar o`GetWirelessDevice`Operação da API ou o`get-wireless-device`Comando da CLI.

Next steps (Próximas etapas)

Depois de configurar os parâmetros, você pode criar grupos multicast e tarefas FUOTA para enviar carga de downlink ou atualizar o firmware dos seus dispositivos LoraWAN.

- Para obter informações sobre a criação de grupos multicast, consulteCrie grupos multicast e adicione dispositivos ao grupo (p. 1183).
- Para obter informações sobre a criação de tarefas do FUOTA, consulte.Crie a tarefa FUOTA e forneça imagem de firmware (p. 1194).

Crie grupos multicast e adicione dispositivos ao grupo

É possível criar grupos multicast usando o console do ou a CLI. Se você estiver criando seu grupo multicast pela primeira vez, recomendamos que você use o console para adicionar seu grupo multicast. Quando você quiser gerenciar seu grupo multicast e adicionar ou remover dispositivos do grupo, você pode usar a CLI.

Depois de trocar a sinalização com os dispositivos finais que você adicionou,AWS IoT Core for LoRaWANestabelece as chaves compartilhadas com os dispositivos finais e configura os parâmetros para a transferência de dados.

Pré-requisitos

Antes de criar grupos multicast e adicionar dispositivos ao grupo:

- Prepare seus dispositivos para configuração multicast e FUOTA especificando os parâmetros de configuração FUOTA`GenAppKeyeFPorts`. Para obter mais informações, consultePreparar dispositivos para configuração multicast e FUOTA (p. 1181)
- Verifique se os dispositivos suportam os modos de operação classe B ou classe C. Dependendo da classe de dispositivo compatível com o dispositivo, escolha um perfil de dispositivo que tenha um ou ambosSuporta classe BouSuporta Classe Cmodos ativados. Para obter informações sobre perfis de dispositivos, consulteAdicionar perfis aoAWS IoT Core for LoRaWAN (p. 1141).

  No início da sessão multicast, uma janela de distribuição de classe B ou classe C é usada para enviar mensagens de downlink para os dispositivos do seu grupo.

Crie grupos de multicast usando o console

Para criar grupos de multicast usando o console do, acesse oGrupos de multicastPágina doAWS IoTConsole do e escolhaCriar grupo multicast.

1. Criar um grupo multicast

   Para criar seu grupo multicast, especifique as propriedades e tags multicast para o grupo.

   1. Especificar propriedades multicast

      Para especificar propriedades multicast, insira as seguintes informações para o grupo multicast.
      - Name (Nome): Digite um nome exclusivo para seu grupo de multicast. O nome deve conter apenas letras, números, hifens e sublinhados. Ele nome não pode conter espaços.
      - Descrição: É possível fornecer uma descrição opcional para o grupo multicast. O comprimento da descrição pode ser de até 2.048 caracteres
   2. Tags para grupo multicast

      Opcionalmente, é possível fornecer pares de chave-valor comoTagspara seu grupo multicast. Para continuar criando seu grupo multicast, escolhaPróximo.
2. Adicionar dispositivos a um grupo de multicast

   Você pode adicionar dispositivos individuais ou um grupo de dispositivos ao seu grupo multicast. Para adicionar dispositivos:

   1. Especificar RfRegion

      Especifique oRfRegionou banda de frequência para seu grupo multicast. ORfRegionpara o seu grupo multicast deve corresponder aoRfRegionDe dispositivos que você adiciona ao grupo multicast. Para obter mais informações sobre aRfRegionConsulte, consulteConsidere a seleção de LoRa bandas de frequência para seus gateways e conexão de dispositivos (p. 1133).
   2. Selecione uma classe de dispositivo multicast

      Escolha se você deseja que os dispositivos no grupo multicast alternem para um modo de classe B ou classe C no início da sessão multicast. Uma sessão de classe B pode receber mensagens de downlink em slots de downlink regulares e uma sessão de classe C pode receber mensagens de downlink a qualquer momento.
   3. Escolha os dispositivos que você deseja adicionar ao grupo

Escolha se deseja adicionar dispositivos individualmente ou em massa ao grupo multicast.

- Para adicionar dispositivos individualmente, insira o ID do dispositivo sem fio de cada dispositivo que você deseja adicionar ao grupo.
- Para adicionar dispositivos em massa, você pode filtrar os dispositivos que deseja adicionar por perfil ou tags do dispositivo. Para o perfil do dispositivo, você pode adicionar dispositivos com um perfil compatível com classe B, classe C ou ambas as classes de dispositivo.

4. Para criar um grupo multicast, selecioneCriar.

   Os detalhes do grupo multicast e os dispositivos que você adicionouaparecem no grupo. Para obter informações sobre o status do grupo multicast e seus dispositivos e para solucionar problemas, consulteMonitore e solucione problemas de status de seu grupo multicast e dispositivos no grupo (p. 1187).

Depois de criar um grupo multicast, você pode escolherAçãopara editar, excluir ou adicionar dispositivos ao grupo multicast. Depois de adicionar os dispositivos, você pode agendar uma sessão para que a carga de downlink seja enviada para os dispositivos do seu grupo.

Crie grupos multicast usando a API

Para criar grupos multicast e adicionar dispositivos ao grupo usando a API:

1. Criar um grupo multicast

   Para criar seu grupo multicast, use o`CreateMulticastGroup`Operação da API ou o`create-multicast-group`Comando da CLI. Você pode fornecer um`input.json`arquivo como entrada para o`create-multicast-group`comando.

```
aws iotwireless create-multicast-group \
    --cli-input-json file://input.json
```

   em que:

   Conteúdo da input.json

```
{
   "Description": "Multicast group to send downlink payload and perform FUOTA
 updates.",
   "LoRaWAN": {
      "DlClass": "ClassB",
      "RfRegion": "US915"
   },
   "Name": "MC_group_FUOTA"
}
```

   Depois de criar seu grupo multicast, você pode usar as seguintes operações de API ou comandos da CLI para atualizar, excluir ou obter informações sobre seus grupos multicast.

   - `UpdateMulticastGroup` ou `update-multicast-group`
   - `GetMulticastGroup` ou `get-multicast-group`
   - `ListMulticastGroups` ou `list-multicast-groups`
   - `DeleteMulticastGroup` ou `delete-multicast-group`

2. Adicionar dispositivos a um grupo de multicast

   É possível adicionar dispositivos ao grupo multicast individualmente ou em massa.

- Para adicionar dispositivos em massa ao seu grupo multicast, use o`StartBulkAssociateWirelessDeviceWithMulticastGroup`Operação da API ou o`start-bulk-associate-wireless-device-with-multicast-group`Comando da CLI. Para filtrar os dispositivos que você deseja associar em massa ao grupo multicast, forneça uma string de consulta. A seguir mostra como você pode adicionar um grupo de dispositivos que tem um perfil de dispositivo com o ID especificado vinculado a ele.

```
aws iotwireless start-bulk-associate-wireless-device-with-multicast-group \
    --id "12abd34e-5f67-89c2-9293-593b1bd862e0" \
    --cli-input-json file://input.json
```

  em que:

  Conteúdo da input.json

```
{
     "QueryString": "DeviceProfileName: MyWirelessDevice AND DeviceProfileId:
 d6d8ef8e-7045-496d-b3f4-ebcaa1d564bf",
     "Tags": [
        {
            "Key": "Multicast",
            "Value": "ClassB"
        }
    ]
}
```

  Aqui,`multicast-groups/d6d8ef8e-7045-496d-b3f4-ebcaa1d564bf/bulk`é o URL usado para associar dispositivos ao grupo.

- Para adicionar dispositivos individualmente ao seu grupo multicast, use o`AssociateWirelessDeviceWithMulticastGroup`Operação da API ou o`associate-wireless-device-with-multicast-group`CLI. Forneça o ID do dispositivo sem fio para cada dispositivo que você deseja adicionar ao grupo.

```
aws iotwireless associate-wireless-device-with-multicast-group \
    --id "12abd34e-5f67-89c2-9293-593b1bd862e0" \
    --wireless-device-id "ab0c23d3-b001-45ef-6a01-2bc3de4f5333"
```

Depois de criar seu grupo multicast, você pode usar as seguintes operações de API ou comandos da CLI para obter informações sobre seu grupo multicast ou para desassociar dispositivos.

- `DisassociateWirelessDeviceFromMulticastGroup` ou `disassociate-wireless-device-from-multicast-group`
- `StartBulkDisassociateWirelessDeviceFromMulticastGroup` ou `start-bulk-disassociate-wireless-device-from-multicast-group`
- `ListWirelessDevices` ou `list-wireless-devices`

  Note

  O`ListWirelessDevices`A operação da API pode ser usada para listar dispositivos sem fio em geral e dispositivos sem fio associados a um grupo multicast ou a uma tarefa FUOTA.

  - Para listar dispositivos sem fio associados a um grupo de multicast, use o`ListWirelessDevices`Operação da API do`MulticastGroupID`como o filtro.
  - Para listar dispositivos sem fio associados a uma tarefa FUOTA, use o`ListWirelessDevices`Operação da API do`FuotaTaskID`como o filtro.

Next steps (Próximas etapas)

Depois de criar um grupo multicast e adicionar dispositivos, você pode continuar adicionando dispositivos e monitorar o status do grupo multicast e seus dispositivos. Se seus dispositivos tiverem sido adicionados com sucesso ao grupo, você poderá configurar e agendar uma mensagem de downlink para ser enviada aos dispositivos. Antes que você possa enviar uma mensagem de downlink, o status de seus dispositivos deve serPronto para configuração multicast. Depois que você agendar uma mensagem de downlink, o status mudará paraTentativa de sessão. Para obter mais informações, consulteAgende uma mensagem de downlink para enviar para dispositivos em seu grupo multicast (p. 1189)

Se você quiser atualizar o firmware dos dispositivos no grupo multicast, você pode executar Atualizações de Firmware Over-The-Air (FUOTA) comAWS IoT Core for LoRaWAN. Para obter mais informações, consulteAtualizações de firmware over-the-air (FUOTA) paraAWS IoT Core for LoRaWANdispositivos (p. 1191)

Se seus dispositivos não foram adicionados ou se você vir um erro no status do grupo multicast ou do dispositivo, poderá passar o mouse sobre o erro para obter mais informações e resolvê-lo. Se você ainda vir um erro, para obter informações sobre como solucionar problemas e resolver o problema, consulteMonitore e solucione problemas de status de seu grupo multicast e dispositivos no grupo (p. 1187).

Monitore e solucione problemas de status de seu grupo multicast e dispositivos no grupo

Depois de adicionar dispositivos e criar seu grupo de multicast, abra aAWS Management Console. Navegue até o .Grupos de multicastPágina doAWS IoTConsole e escolha o grupo multicast que você criou para exibir seus detalhes. Você verá informações sobre o grupo multicast, o número de dispositivos que foram adicionados edispositivodetalhes de status. Você pode usar as informações de status para rastrear o andamento da sua sessão multicast e solucionar problemas de erros.

Status do grupo multicast

O grupo multicast pode ter uma das mensagens de status a seguir exibidas naAWS Management Console.

- Pendente

  Esse status indica que você criou um grupo multicast, mas ele ainda não tem uma sessão multicast. Você verá essa mensagem de status exibida quando seu grupo for criado. Durante esse período, você pode atualizar seu grupo multicast e associar ou desassociar dispositivos ao grupo. Depois que o status mudar dePendente, dispositivos adicionais não podem seradicionadapara o grupo.
- Tentativa de sessão

  Depois que seus dispositivos tiverem sido adicionados com êxito ao grupo multicast, quando o grupo tiver uma sessão multicast agendada, você verá essa mensagem de status exibida. Durante esse período, não será possível atualizar nem adicionar dispositivos ao grupo de multicast. Se você cancelar sua sessão multicast, o status do grupo mudará paraPendente.
- Na sessão

  Quando for a primeira hora da sessãopara sua sessão multicast, você verá essa mensagem de status exibida. Um grupo multicast também continua nesse estado quando está associado a uma tarefa FUOTA que tem uma sessão de atualização de firmware em andamento.

  Se você não tiver uma tarefa FUOTA associada na sessão e se a sessão multicast for cancelada porque o tempo da sessão excedeu o tempo limite ou você cancelou sua sessão multicast, o status do grupo mudará paraPendente.
- Excluir espera

Se você excluir seu grupo multicast, o status do grupo será alterado paraExcluir espera. As exclusões são permanentes e não podem ser desfeitas. Essa ação pode levar tempo e o status do grupo seráDelete_WaitingAté que o grupo multicast tenha sido excluído. Depois que seu grupo multicast entrar nesse estado, ele não poderá fazer a transição para um dos outros estados.

Status dos dispositivos no grupo multicast

Os dispositivos em seu grupo multicast podem ter uma das seguintes mensagens de status exibidas naAWS Management Console. Você pode passar o mouse sobre cada mensagem de status para obter mais informações sobre o que ela indica.

- Package de tentativas

  Depois que seus dispositivos tiverem sido associados ao grupo multicast, o status do dispositivo seráPackage de tentativas. Esse status indica queAWS IoT Core for LoRaWANainda não confirmou se o dispositivo suporta configuração e operação multicast.
- Package sem suporte

  Depois que seus dispositivos tiverem sido associados ao grupo multicast,AWS IoT Core for LoRaWANverifica se o firmware do dispositivo é capaz de configurar e operar multicast. Se o dispositivo não tiver o pacote multicast suportado, seu status seráPackage sem suporte. Para resolver o erro, verifique se o firmware do dispositivo é capaz de configurar e operar multicast.
- Tentativa de configuração de multicast

  Se os dispositivos associados ao seu grupo multicast forem capazes de configuração e operação de multicast, o status seráTentativa de configuração de multicast. Esse status indica que o dispositivo ainda não concluiu a configuração de multicast.
- Pronto para configuração multicast

  Seu dispositivo concluiu a configuração de multicast e foi adicionado ao grupo multicast. Esse status indica que os dispositivos estão prontos para uma sessão multicast e uma mensagem de downlink pode ser enviada para esses dispositivos. O status também indica quando você pode usar o FUOTA para atualizar o firmware dos dispositivos no grupo.
- Tentativa de sessão

  Uma sessão multicast foi agendada para os dispositivos em seu grupo multicast. No início de uma sessão de grupo multicast, o status do dispositivo éTentativa de sessãoe solicitaçõessão enviadospara saber se uma janela de distribuição de classe B ou classe C pode ser iniciada para a sessão. Se o tempo necessário para configurar a sessão multicast exceder o tempo limite ou se você cancelar a sessão multicast, o status mudará paraConfiguração de multicast.
- Na sessão

  Esse status indica que uma janela de distribuição de classe B ou classe C foi iniciada e seu dispositivo tem uma sessão multicast em andamento. Durante esse período, mensagens de downlink podem ser enviadas deAWS IoT Core for LoRaWANPara dispositivos no grupo multicast. Se você atualizar a hora da sessão, ela substituirá a sessão atual e o status mudará paraTentativa de sessão. Quando o tempo da sessão termina ou se você cancelar a sessão multicast, o status muda paraPronto para configuração multicast.

Next steps (Próximas etapas)

Agora que você aprendeu os diferentes status do seu grupo multicast e dos dispositivos em seu grupo, e como solucionar problemas, como quando um dispositivo não é capaz de configurar multicast, você pode agendar uma mensagem de downlink para ser enviada aos dispositivos e seu grupo multicast estará emNa

sessão. Para obter informações sobre como programar uma mensagem de downlink, consulteAgende uma mensagem de downlink para enviar para dispositivos em seu grupo multicast (p. 1189).

Agende uma mensagem de downlink para enviar para dispositivos em seu grupo multicast

Depois de adicionar dispositivos ao grupo multicast com êxito, você pode iniciar uma sessão multicast e configurar uma mensagem de downlink para ser enviada para esses dispositivos. A mensagem de downlink deve ser agendadaEm 48 horas ae a hora de início para o multicast deve estar emmenos 30 minutos depois da hora atual.

> Note
>
> Dispositivos em um grupo multicast não podem confirmar quando uma mensagem de downlink foi recebida.

Pré-requisitos

Antes de enviar uma mensagem de downlink, você deve ter criado um grupo multicast e adicionar dispositivos com êxito ao grupo para o qual deseja enviar uma mensagem de downlink. Você não pode adicionar mais dispositivos após uma hora de início ter sido agendada para sua sessão multicast. Para obter mais informações, consulteCrie grupos multicast e adicione dispositivos ao grupo (p. 1183)

Se algum dos dispositivos não tiver sido adicionado com sucesso, o grupo multicast e o status do dispositivo conterão informações para ajudá-lo a resolver os erros. Se os erros ainda persistirem, para obter informações sobre a solução de problemas desses erros, consulteMonitore e solucione problemas de status de seu grupo multicast e dispositivos no grupo (p. 1187).

Agende uma mensagem de downlink usando o console

Para enviar uma mensagem de downlink usando o console do, acesse oGrupos de multicastPágina doAWS IoTconsole e escolha o grupo multicast que você criou. Na página de detalhes do grupo multicast, selecioneAgendar mensagem de downlinke, depois, escolhaAgendar sessão de downlink.

1. Janela de mensagem de downlink agendado

   Você pode configurar uma janela de tempo para que uma mensagem de downlink seja enviada para os dispositivos em seu grupo multicast. A mensagem de downlink deve ser agendada dentro de 48 horas.

   Para programar sua sessão multicast, especifique os parâmetros a seguir:

   - Data de inícioeHorário de início: A data e a hora de início devem ser pelo menos 30 minutos após e 48 horas antes da hora atual.

     > Note
     >
     > A hora especificada é em UTC, então considere verificar a diferença de horário com seu fuso horário ao agendar a janela de downlink.

   - Tempo limite de sessão: O tempo após o qual você deseja que a sessão multicast atinja o tempo limite se nenhuma mensagem de downlink tiver sido recebida. O tempo limite mínimo permitido é de 60 segundos. O valor máximo de tempo limite é de 2 dias para grupos multicast da classe B e 18 horas para grupos multicast da classe C.

2. Configurar sua mensagem de downlink

   Para configurar sua mensagem de downlink, especifique os parâmetros a seguir:

- Taxa de dados: Escolha uma taxa de dados para sua mensagem de downlink. A taxa de dados depende do tamanho do RFregion e da carga útil. A taxa de dados padrão é 8 para a região US915 e 0 para a região EU868.
- Frequência: Escolha uma frequência para enviar sua mensagem de downlink. Para evitar conflitos de mensagens, escolha uma frequência disponível dependendo do RFregion.
- FPort: Escolha uma porta de frequência disponível para enviar a mensagem de downlink para seus dispositivos.
- Carga útil: Especifique o tamanho máximo da carga útil, dependendo da taxa de dados. Usando a taxa de dados padrão, você pode ter um tamanho máximo de carga útil de 33 bytes no US915 RfRegion e 51 bytes na EU868 RFregion. Usando taxas de dados maiores, você pode transferir até um tamanho máximo de carga útil de 242 bytes.

Para agendar sua mensagem de downlink, escolhaSchedule.

Agende uma mensagem de downlink usando a API

Para agendar uma mensagem de downlink usando a API, use o`StartMulticastGroupSession`Operação da API ou o`start-multicast-group-session`Comando da CLI.

Você pode usar as seguintes operações de API ou comandos da CLI para obter informações sobre um grupo multicast e excluir um grupo multicast.

- `GetMulticastGroupSession` ou `get-multicast-group-session`
- `DeleteMulticastGroupSession` ou `delete-multicast-group-session`

Para enviar dados para um grupo multicast após a sessão ter sido iniciada, use o`SendDataToMulticastGroup`Operação da API ou o`send-data-to-multicast-group`Comando da CLI.

Next steps (Próximas etapas)

Depois de configurar uma mensagem de downlink para ser enviada aos dispositivos, a mensagem é enviada no início da sessão. Os dispositivos em um grupo multicast não podem confirmar se a mensagem foi recebida.

Configurar mensagens de downlink adicionais

Você também pode configurar mensagens de downlink adicionais para serem enviadas aos dispositivos em seu grupo multicast:

- Para configurar mensagens de downlink adicionais a partir do console:
  1. Vá para oGrupos de multicastPágina doAWS IoTconsole e escolha o grupo multicast que você criou.
  2. Na página de detalhes do grupo multicast, selecioneAgendar mensagem de downlinke, depois, escolhaConfigurar mensagem de downlink adicional.
  3. Especificar os parâmetrosTaxa de dados,Frequência,FPort, eCarga útil, semelhante a como você configurou esses parâmetros para sua primeira mensagem de downlink.
- Para configurar mensagens de downlink adicionais usando a API ou a CLI, chame a`SendDataToMulticastGroup`Operação da API ou o`send-data-to-multicast-group`Comando da CLI para cada mensagem de downlink adicional.

Atualizar a programação da sessão

Você também pode atualizar a programação da sessão para usar uma nova data e hora de início para sua sessão multicast. O novo cronograma de sessão substituirá a sessão agendada anteriormente.

> **Note**
>
> Atualize sua sessão multicast somente quando necessário. Essas atualizações podem fazer com que um grupo de dispositivos acorde por um longo período e drene a bateria.

- Para atualizar a programação da sessão a partir do console:
  1. Vá para oGrupos de multicastPágina doAWS IoTconsole e escolha o grupo multicast que você criou.
  2. Na página de detalhes do grupo multicast, selecioneAgendar mensagem de downlinke, depois, escolhaAtualizar a programação da sessão.
  3. Especificar os parâmetrosData do estado,Horário de início, eTempo limite de sessão, semelhante a como você especificou esses parâmetros para sua primeira mensagem de downlink.
- Para atualizar o cronograma de sessão a partir da API ou da CLI, use o`StartMulticastGroupSession`Operação da API ou o`start-multicast-group-session`Comando da CLI.

Atualizações de firmware over-the-air (FUOTA) paraAWS IoT Core for LoRaWANdispositivos

Use Atualizações de Firmware Over-The-Air (FUOTA) para implantar atualizações de firmware paraAWS IoT Core for LoRaWANDispositivos.

Usando o FUOTA, você pode enviar atualizações de firmware para dispositivos individuais ou para um grupo de dispositivos. Você também pode enviar atualizações de firmware para vários dispositivos criando um grupo multicast. Primeiro adicione seus dispositivos ao grupo multicast e, em seguida, envie sua imagem de atualização de firmware para todos esses dispositivos. Recomendamos que você assine digitalmente as imagens de firmware para que os dispositivos que recebem as imagens possam verificar se estão vindo da fonte correta.

comAWS IoT Core for LoRaWANatualizações FUOTA, você pode:

- Implante novas imagens de firmware ou imagens delta em um único dispositivo ou em um grupo de dispositivos.
- Verifique a autenticidade e a integridade do novo firmware depois de implantá-lo nos dispositivos.
- Monitore o andamento de problemas de implantação e depuração em caso de falha na implantação.

AWS IoT Core for LoRaWANo suporte do FUOTA e grupos multicast é baseado noAliança LoRaseguintes especificações:

- Especificação de configuração de multicast remoto LoRaWAN, TS005-2.0.0
- Especificação de transporte de bloco de dados fragmentados LoRaWAN, TS004-2.0.0
- Especificação de sincronização do relógio da camada de aplicativo LoRaWAN, TS003-2.0.0

> **Note**
>
> AWS IoT Core for LoRaWANexecuta automaticamente a sincronização do relógio de acordo com o LoRa Especificação da aliança. Ele usa a função`AppTimeReq`para responder a hora do lado do servidor aos dispositivos que o solicitam usando a sinalização ClockSync.

Veja a seguir como executar atualizações do FUOTA.

- Visão geral do processo FUOTA (p. 1192)
- Crie a tarefa FUOTA e forneça imagem de firmware (p. 1194)
- Adicionardispositivos e grupos multicast para uma tarefa FUOTA e agende uma sessão FUOTA (p. 1196)
- Monitore solucionar problemas do status da tarefa e dos dispositivos FUOTA adicionados à tarefa (p. 1198)

Visão geral do processo FUOTA

O diagrama a seguir mostra comoAWS IoT Core for LoRaWANexecuta o processo FUOTA para seus dispositivos finais. Se você estiver adicionando dispositivos individuais à sua sessão FUOTA, poderá ignorar as etapas para criar e configurar seu grupo multicast. Você pode adicionar seus dispositivos diretamente a uma sessão FUOTA eAWS IoT Core for LoRaWANiniciará o processo de atualização do firmware.

Para executar atualizações FUOTA para seus dispositivos, primeiro crie sua imagem de firmware assinada digitalmente e configure os dispositivos e grupos multicast que você deseja adicionar à sua tarefa FUOTA. Depois de iniciar uma sessão FUOTA, seus dispositivos finais coletam todos os fragmentos, reconstroem a imagem dos fragmentos, relatam o status paraAWS IoT Core for LoRaWANE, em seguida, aplique a nova imagem de firmware.

O seguinte ilustra as diferentes etapas no processo FUOTA:

1. Crie uma imagem de firmware ou imagem delta com uma assinatura digital

   para oAWS IoT Core for LoRaWANpara executar atualizações FUOTA para seus dispositivos LoraWAN, recomendamos que você assine digitalmente a imagem de firmware ou a imagem delta ao enviar atualizações de firmware pelo ar. Os dispositivos que recebem as imagens podem verificar se elas vêm da fonte correta.

   Sua imagem de firmware não deve ter tamanho maior que 1 megabyte. Quanto maior o tamanho do firmware, mais tempo pode levar para o processo de atualização ser concluído. Para uma transferência de dados mais rápida ou se sua nova imagem for maior que 1 Megabyte, use uma imagem delta, que é a parte da nova imagem que é o delta entre sua nova imagem de firmware e a imagem anterior.

> **Note**
>
> AWS IoT Core for LoRaWANnão fornece a ferramenta de geração de assinatura digital e o sistema de gerenciamento de versão do firmware. Você pode usar qualquer ferramenta de terceiros para gerar a assinatura digital para sua imagem de firmware. Recomendamos que você use uma ferramenta de assinatura digital, como a incorporada noARM Mbed GitHub repositório, que também inclui ferramentas para gerar a imagem delta e para dispositivos usarem essa imagem.

2. Identificar e configurar os dispositivos para FUOTA

   Depois de identificar os dispositivos do FUOTA, envie atualizações de firmware para dispositivos individuais ou múltiplos.

   - Para enviar atualizações de firmware para vários dispositivos, crie um grupo multicast e configure o grupo multicast com dispositivos finais. Para obter mais informações, consulteCrie grupos multicast para enviar uma carga de downlink para vários dispositivos (p. 1180)
   - Para enviar atualizações de firmware para dispositivos individuais, adicione esses dispositivos à sua sessão FUOTA e, em seguida, execute a atualização do firmware.

3. Programe uma janela de distribuição e configure a sessão de fragmentação

   Se você criou um grupo multicast, poderá especificar a janela de distribuição classe B ou classe C para determinar quando os dispositivos podem receber os fragmentos deAWS IoT Core for LoRaWAN. Seus dispositivos podem estar operando na classe A antes de mudarem para o modo classe B ou classe C. Você também deve especificar a hora de início da sessão.

   Os dispositivos de classe B ou classe C ativam na janela de distribuição especificada e começam a receber os pacotes de downlink. Os dispositivos que operam no modo classe C podem consumir mais energia do que os dispositivos de classe B. Para obter mais informações, consulteClasses de dispositivos (p. 1174)

4. Dispositivos finais relatam status paraAWS IoT Core for LoRaWANe atualize a imagem do firmware

   Depois de configurar uma sessão de fragmentação, seus dispositivos finais eAWS IoT Core for LoRaWANExecute as etapas a seguir para atualizar o firmware dos seus dispositivos.

   1. Como os dispositivos LoraWAN têm uma baixa taxa de dados, para iniciar o processo FUOTA,AWS IoT Core for LoRaWANconfigura uma sessão de fragmentação para fragmentar a imagem do firmware. Entãoele envia esses fragmentos para os dispositivos finais.
   2. DepoisAWS IoT Core for LoRaWANenvia os fragmentos de imagem, seus dispositivos finais LoRaWAN executam as seguintes tarefas.
      a. Colete os fragmentos e reconstrua a imagem binária desses fragmentos.
      b. Verifique a assinatura digital da imagem reconstruída para autenticar a imagem e verificar se ela vem da fonte correta.
      c. Compare a versão do firmware deAWS IoT Core for LoRaWANPara a versão atual.
      d. Relate o status das imagens fragmentadas que foram transferidas paraAWS IoT Core for LoRaWANE, em seguida, aplique a nova imagem de firmware.

      > **Note**
      >
      > Em alguns casos,os dispositivos finais relatam o status das imagens fragmentadas que foram transferidas paraAWS IoT Core for LoRaWANAntes de verificar a assinatura digital da imagem do firmware.

Agora que você aprendeu o processo FUOTA, você pode criar sua tarefa FUOTA e adicionar dispositivos à tarefa para atualizar o firmware. Para obter mais informações, consulteCrie a tarefa FUOTA e forneça imagem de firmware (p. 1194)

Crie a tarefa FUOTA e forneça imagem de firmware

Para atualizar o firmware dos seus dispositivos LoraWAN, primeiro crie uma tarefa FUOTA e forneça a imagem de firmware assinada digitalmente que você deseja usar para a atualização. Em seguida, você pode adicionar seus dispositivos e grupos multicast à tarefa e agendar uma sessão FUOTA. Quando a sessão começa,AWS IoT Core for LoRaWANconfigura uma sessão de fragmentação e seus dispositivos finais coletam os fragmentos, reconstroem a imagem e aplicam o novo firmware. Para obter informações sobre o processo FUOTA, consulte.Visão geral do processo FUOTA (p. 1192).

A seguir mostra como você pode criar uma tarefa FUOTA e carregar a imagem de firmware ou a imagem delta que você armazenará em um bucket do S3.

Pré-requisitos

Antes que você possa executar atualizações do FUOTA, a imagem do firmware deve ser assinada digitalmente para que seus dispositivos finais possam verificar a autenticidade da imagem ao aplicar a imagem. Você pode usar qualquer ferramenta de terceiros para gerar a assinatura digital para sua imagem de firmware. Recomendamos que você use uma ferramenta de assinatura digital, como a incorporada noRepositório do ARM Mbed GitHub, que também inclui ferramentas para gerar a imagem delta e para dispositivos usarem essa imagem.

Crie uma tarefa FUOTA e carregue a imagem de firmware usando o console do

Para criar uma tarefa FUOTA e carregar sua imagem de firmware usando o console, vá para aTarefas FUOTAguia do console e, em seguida, escolhaCriar tarefa do FUOTA.

1. Criar tarefa do FUOTA

   Para criar sua tarefa FUOTA, especifique as propriedades e as tags da tarefa.

   1. Especifique as propriedades da tarefa FUOTA

      Para especificar as propriedades da tarefa FUOTA, insira as seguintes informações para sua tarefa FUOTA.

      - Name (Nome): Digite um nome exclusivo para sua tarefa FUOTA. O nome deve conter apenas letras, números, hifens e sublinhados. Ele nome não pode conter espaços.
      - Descrição: É possível fornecer uma descrição opcional para o grupo multicast. O campo de descrição pode ter até 2.048 caracteres
      - RfRegion: Defina a faixa de frequência para sua tarefa FUOTA. A banda de frequência deve corresponder à que você usou para provisionar seus dispositivos sem fio ou grupos multicast.

   2. Etiquetas para FUOTA task

      Opcionalmente, é possível fornecer pares de chave-valor comoTagspara sua tarefa FUOTA. Para continuar criando sua tarefa, escolhaPróximo.

2. Carregar imagem de firmware

   Escolha o arquivo de imagem de firmware que você deseja usar para atualizar o firmware dos dispositivos adicionados à tarefa FUOTA. O arquivo de imagem de firmware é armazenado em um bucket do S3. Você pode fornecerAWS IoT Core for LoRaWANAs permissões para acessar a imagem do firmware em seu nome. Recomendamos que você assine digitalmente as imagens de firmware para que sua autenticidade seja verificada quando a atualização do firmware for executada.

   1. Escolha o arquivo de imagem de firmware

      Você pode carregar um novo arquivo de imagem de firmware para um bucket do S3 ou escolher uma imagem existente que já tenha sido carregada em um bucket do S3.

Note

O arquivo de imagem do firmware não deve ter tamanho maior que 1 megabyte. Quanto maior o tamanho do firmware, mais tempo pode levar para o processo de atualização ser concluído.

- Para usar uma imagem existente, selecioneSelecione uma imagem de firmware existente, escolhaNavegar no S3E, em seguida, o arquivo de imagem de firmware que você deseja usar.

  AWS IoT Core for LoRaWANPreenche o URL do S3, que é o caminho para o arquivo de imagem do firmware no bucket do S3. O formato do caminho dos`s3://bucket_name/file_name`. Para visualizar o arquivo noAmazon Simple Storage ServiceConsole do, escolhaExibir.
- Para carregar uma nova imagem de firmware.
  a. SelecioneCarregar uma nova imagem de firmwareumaE faça o upload da imagem de firmware. O arquivo de imagem não deve ser maior que 1 megabyte.
  b. Para criar um bucket do S3 e inserir umBucket namepara armazenar seu arquivo de imagem de firmware, escolhaCriar bucket do S3.

2. Permissões para acessar o bucket

   É possível criar uma nova função de serviço ou escolher uma função existente para permitirAWS IoT Core for LoRaWANPara acessar o arquivo de imagem de firmware no bucket do S3 em seu nome. Escolha Next (Próximo).

   Para criar uma nova função, você pode inserir um nome de função ou deixá-lo em branco para que um nome aleatório seja gerado automaticamente. Para exibir as permissões de política que concedem acesso ao bucket do S3, escolhaExibir permissões da política.

Para obter mais informações sobre como usar um bucket do S3 para armazenar sua imagem e concessãoAWS IoT Core for LoRaWANpermissões para acessá-lo, consulteFaça upload do arquivo de firmware em um bucket do S3 e adicione uma função do IAM (p. 1166).

3. Analisar e criar

   Para criar sua tarefa FUOTA, revise a tarefa FUOTA e os detalhes de configuração que você especificou e escolhaPágina Create task.

Crie a tarefa FUOTA e carregue a imagem de firmware usando a API

Para criar uma tarefa FUOTA e especificar seu arquivo de imagem de firmware usando a API, use o`CreateFuotaTask`Operação da API ou o`create-fuota-task`Comando da CLI. Você pode fornecer um`input.json`arquivo como entrada para o`create-fuota-task`comando. Quando você usa a API ou a CLI, o arquivo de imagem de firmware fornecido como entrada já deve estar carregado em um bucket do S3. Você também especifica a função do IAM que forneceAWS IoT Core for LoRaWANAcesso à imagem de firmware no bucket do S3.

```
aws iotwireless create-fuota-task \
    --cli-input-json file://input.json
```

em que:

Conteúdo da input.json

```
{
   "Description": "FUOTA task to update firmware of devices in multicast group.",
   "FirmwareUpdateImage": "S3:/firmware_bucket/firmware_image
   "FirmwareUpdateRole": "arn:aws:iam::123456789012:role/service-role/ACF1zBEI"
```

```
    "LoRaWAN": {
       "RfRegion": "US915"
    },
    "Name": "FUOTA_Task_MC"
}
```

Depois de criar sua tarefa FUOTA, você pode usar as seguintes operações de API ou comandos da CLI para atualizar, excluir ou obter informações sobre sua tarefa FUOTA.

- `UpdateFuotaTask` ou `update-fuota-task`
- `GetFuotaTask` ou `get-fuota-task`
- `ListFuotaTasks` ou `list-fuota-tasks`
- `DeleteFuotaTask` ou `delete-fuota-task`

Next steps (Próximas etapas)

Agora que você criou uma tarefa FUOTA e forneceu a imagem do firmware, você pode adicionar dispositivos à tarefa para atualizar o firmware. Você pode adicionar dispositivos individuais ou grupos multicast à tarefa. Para obter mais informações, consulteAdicionardispositivos e grupos multicast para uma tarefa FUOTA e agende uma sessão FUOTA (p. 1196)

Adicionardispositivos e grupos multicast para uma tarefa FUOTA e agende uma sessão FUOTA

Depois de criar uma tarefa FUOTA, você pode adicionar dispositivos à sua tarefa para a qual deseja atualizar o firmware. Depois que seus dispositivos tiverem sido adicionados com sucesso à tarefa FUOTA, você poderá agendar uma sessão FUOTA para atualizar o firmware do dispositivo.

- Se você tiver apenas um pequeno número de dispositivos, poderá adicioná-los diretamente à sua tarefa FUOTA.
- Se você tiver um grande número de dispositivosPara o qual você deseja atualizar o firmware para, você pode adicionar esses dispositivos aos seus grupos multicast e, em seguida, adicionar os grupos multicast à sua tarefa FUOTA. Para obter informações sobre a criação e o uso de grupos multicast, consulteCrie grupos multicast para enviar uma carga de downlink para vários dispositivos (p. 1180).

> **Note**
>
> Você pode adicionar dispositivos individuais ou grupos multicast à tarefa FUOTA. Você não pode adicionar dispositivos e grupos multicast à tarefa.

Depois de adicionar seus dispositivos ou grupos multicast, você pode iniciar uma sessão de atualização de firmware.AWS IoT Core for LoRaWANcoleta a imagem do firmware, fragmenta as imagens e armazena os fragmentos em um formato criptografado. Seus dispositivos finais coletam os fragmentos e aplicam a nova imagem de firmware. O tempo que leva para a atualização do firmware depende do tamanho da imagem e de como as imagens foram fragmentadas. Após a conclusão da atualização do firmware, os fragmentos criptografados da imagem do firmware armazenados porAWS IoT Core for LoRaWANé excluído. Você ainda pode encontrar a imagem de firmware no bucket do S3.

Pré-requisitos

Antes de adicionar dispositivos ou grupos multicast à sua tarefa FUOTA, faça o seguinte.

- Você já deve ter criado a tarefa FUOTA e fornecido sua imagem de firmware. Para obter mais informações, consulteCrie a tarefa FUOTA e forneça imagem de firmware (p. 1194)

- Provisione os dispositivos sem fio para os quais você deseja atualizar o firmware do dispositivo. Para obter mais informações sobre como integrar seu dispositivo, consulteIncorporar seus dispositivosAWS IoT Core for LoRaWAN (p. 1139).
- Para atualizar o firmware de vários dispositivos, você pode adicioná-los a um grupo multicast. Para obter mais informações, consulteCrie grupos multicast para enviar uma carga de downlink para vários dispositivos (p. 1180)
- Quando você integra os dispositivos paraAWS IoT Core for LoRaWAN, especifique o parâmetro de configuração FUOTA`FPorts`. Se você estiver usando um dispositivo LoRawan v1.0.x, você também deve especificar o`GenAppKey`. Para obter mais informações sobre os parâmetros de configuração do FUOTA, consultePreparar dispositivos para configuração multicast e FUOTA (p. 1181).

Adicione dispositivos a uma tarefa FUOTA e agende uma sessão FUOTA usando o console

Para adicionar dispositivos ou grupos multicast e agendar uma sessão FUOTA usando o console, vá para aTarefas FUOTAGuia do console do. Em seguida, escolha a tarefa FUOTA à qual você deseja adicionar dispositivos e execute a atualização do firmware.

Adicionar dispositivos e grupos multicast

1. Você pode adicionar dispositivos individuais ou grupos multicast à sua tarefa FUOTA. No entanto, você não pode adicionar dispositivos individuais e grupos multicast à mesma tarefa FUOTA. Para adicionar dispositivos usando o by console do, siga o procedimento a seguir.

   1. NoDetalhes da tarefa FUOTA, escolhaAdicionar dispositivo.
   2. Escolha a faixa de frequência ouRfRegionpara os dispositivos que você adiciona à tarefa. Esse valor deve corresponder aoRfRegionque você escolheu para a tarefa FUOTA.
   3. Escolha se deseja adicionar dispositivos individuais ou grupos multicast à tarefa.
      - Para adicionar dispositivos individuais, escolhaAdicionar dispositivos individuaise insira o ID do dispositivo de cada dispositivo que você deseja adicionar à sua tarefa FUOTA.
      - Para adicionar grupos multicast, escolhaAdicionar grupos multicaste adicione seus grupos multicast à tarefa. Você pode filtrar os grupos multicast que deseja adicionar à tarefa usando o perfil ou as tags do dispositivo. Ao filtrar por perfil de dispositivo, você pode escolher grupos multicast com dispositivos que têm um perfil comSuporta classe BouSuporta Classe Chabilitado.
2. Programar sessão do FUOTA

   Depois que seus dispositivos ou grupos multicast tiverem sido adicionados com sucesso, você poderá agendar uma sessão FUOTA. Para agendar uma sessão, faça o seguinte.

   1. Escolha a tarefa FUOTA para a qual deseja atualizar o firmware do dispositivo e, em seguida, escolhaProgramar sessão do FUOTA.
   2. Especifique umData de inícioeHorário de início para sua sessão FUOTA. Certifique-se de que a hora de início seja 30 minutos ou mais tarde a partir da hora atual.

Adicione dispositivos a uma tarefa FUOTA e agende uma sessão FUOTA usando a API

Você pode usar oAWS IoT WirelessAPI ou CLI para adicionar seus dispositivos sem fio ou grupos multicast à sua tarefa FUOTA. Em seguida, você pode agendar uma sessão FUOTA.

1. Adicionar dispositivos e grupos multicast

   Você pode associar dispositivos sem fio ou grupos multicast à sua tarefa FUOTA.

- Para associar dispositivos individuais à sua tarefa FUOTA, use o`AssociateWirelessDeviceWithFuotaTask`Operação da API ou o`associate-wireless-device-with-fuota-task`Comando CLI e forneça o`WirelessDeviceID`como entrada.

```
aws iotwireless associate-wireless-device-with-fuota-task \
    --id "01a23cde-5678-4a5b-ab1d-33456808ecb2"
    --wireless-device-id "ab0c23d3-b001-45ef-6a01-2bc3de4f5333"
```

- Para associar grupos multicast à sua tarefa FUOTA, use o`AssociateMulticastGroupWithFuotaTask`Operação da API ou o`associate-multicast-group-with-fuota-task`Comando CLI e forneça o`MulticastGroupID`como entrada.

```
aws iotwireless associate-multicast-group-with-FUOTA-task \
    --id 01a23cde-5678-4a5b-ab1d-33456808ecb2"
    --multicast-group-id
```

Depois de associar seus dispositivos sem fio ou grupo multicast à sua tarefa FUOTA, use as seguintes operações de API ou comandos da CLI para listar seus dispositivos ou grupos multicast ou para desassociá-los da tarefa.

- `DisassociateWirelessDeviceFromFuotaTask` ou `disassociate-wireless-device-from-fuota-task`
- `DisassociateMulticastGroupFromFuotaTask` ou `disassociate-multicast-group-from-fuota-task`
- `ListWirelessDevices` ou `list-wireless-devices`
- `ListMulticastGroups` ou `list-multicast-groups-by-fuota-task`

  Note

  A API:

  - `ListWirelessDevices`pode listar dispositivos sem fio em geral e dispositivos associados a um grupo multicast, quando`MulticastGroupID`é usado como filtro. A API lista dispositivos sem fio associados a uma tarefa FUOTA quando`FuotaTaskID`é usado como filtro.
  - `ListMulticastGroups`pode listar grupos multicast em grupos gerais e multicast associados a uma tarefa FUOTA quando`FuotaTaskID`é usado como filtro.

2. Programar sessão do FUOTA

   Depois que seus dispositivos ou grupos multicast tiverem sido adicionados com sucesso à tarefa FUOTA, você poderá iniciar uma sessão FUOTA para atualizar o firmware do dispositivo. A hora de início deve ser de 30 minutos ou mais tarde a partir da hora atual. Para agendar uma sessão FUOTA usando a API ou a CLI, use o`StartFuotaTask`Operação da API ou o`start-fuota-task`Comando da CLI.

   Depois de iniciar uma sessão FUOTA, você não poderá mais adicionar dispositivos ou grupos multicast à tarefa. Você pode obter informações sobre o status da sessão do FUOTA usando o`GetFuotaTask`Operação da API ou o`get-fuota-task`Comando da CLI.

Monitore solucionar problemas do status da tarefa e dos dispositivos FUOTA adicionados à tarefa

Depois de provisionar os dispositivos sem fio e criar qualquer grupo de multicast que você queira usar, você pode iniciar uma sessão FUOTA executando as etapas a seguir.

Status da tarefa FUOTA

Sua tarefa FUOTA pode ter uma das seguintes mensagens de status exibidas naAWS Management Console.

- Pendente

  Esse status indica que você criou uma tarefa FUOTA, mas ela ainda não tem uma sessão de atualização de firmware. Você verá essa mensagem de status exibida quando sua tarefa for criada. Durante esse período, você pode atualizar sua tarefa FUOTA eassociarou desassocie dispositivos ou grupos multicast com sua tarefa. Depois que o status mudar dePendente, Dispositivos adicionais não podem ser adicionados à tarefa.
- Sessão FUOTA esperando

  Depois que seus dispositivos tiverem sidoadicionadacom êxito para a tarefa FUOTA, quando sua tarefa tiver uma sessão de atualização de firmware agendada, você verá essa mensagem de status exibida. Durante esse período, não será possível atualizar nem adicionar dispositivos à sessão do FUOTA. Se você cancelar sua sessão FUOTA, o status do grupo mudará paraPendente.
- Na sessão do FUOTA

  Quando sua sessão FUOTA começar, você verá essa mensagem de status exibida. A sessão de fragmentação é iniciada e seus dispositivos finais coletam os fragmentos, reconstroem a imagem do firmware, comparam a nova versão do firmware com a versão original e aplicam a nova imagem.
- FUOTA pronto

  Depois que seus dispositivos finais se reportam aAWS IoT Core for LoRaWANque a nova imagem de firmware foi aplicada, ou quando a sessão expirar, a sessão FUOTA está marcada como concluída e você verá esse status exibido.

  Você também verá esse status exibido em qualquer um dos seguintes casos, portanto, verifique se a atualização do firmware foi aplicada corretamente aos dispositivos.
  - Quando o status da tarefa FUOTA eraSessão FUOTA esperandoe há um erro de bucket do S3, como o link para o arquivo de imagem no bucket do S3 está incorreto ouAWS IoT Core for LoRaWANNão tem permissões suficientes para acessar o arquivo no bucket.
  - Quando o status da tarefa FUOTA eraSessão FUOTA esperando, e há uma solicitação para iniciar uma sessão FUOTA, mas uma resposta não é recebida dos dispositivos ou grupos multicast em sua tarefa FUOTA.
  - Quando o status da tarefa FUOTA eraNa sessão do FUOTA, e os dispositivos ou grupos de multicast não enviaram fragmentos por um determinado período de tempo, o que resulta no tempo limite da sessão.
- Excluir espera

  Se você excluir sua tarefa FUOTA que está em qualquer um dos outros estados, você verá esse status exibido. Uma ação de exclusão é permanente e não pode ser desfeita. Essa ação pode levar tempo e o status da tarefa seráExcluir esperaAté que a tarefa FUOTA tenha sido excluída. Depois que sua tarefa FUOTA entrar nesse estado, ela não poderá fazer a transição para um dos outros estados.

Status dos dispositivos em uma tarefa FUOTA

Os dispositivos em sua tarefa FUOTA podem ter uma das seguintes mensagens de status exibidas naAWS Management Console. Você pode passar o mouse sobre cada mensagem de status para obter mais informações sobre o que ela indica.

- Inicial

Quando é a hora de início da sessão do FUOTA,AWS IoT Core for LoRaWANverifica se o dispositivo tem o pacote suportado para a atualização de firmware. Se o dispositivo tiver o pacote suportado, a sessão FUOTA para o dispositivo será iniciada. A imagem do firmware está fragmentada e os fragmentos são enviados para o dispositivo. Quando você vê esse status exibido, ele indica que a sessão FUOTA para o dispositivo ainda não foi iniciada.

- Package sem suporte

  Se o dispositivo não tiver o pacote FUOTA suportado, você verá esse status exibido. Se o pacote de atualização de firmware não for suportado, a sessão FUOTA do seu dispositivo não poderá ser iniciada. Para resolver esse erro, verifique se o firmware do dispositivo pode receber atualizações de firmware usando o FUOTA.

- Algoritmo de fragmentação sem suporte

  No início da sua sessão FUOTA,AWS IoT Core for LoRaWANconfigura uma sessão de fragmentação para o dispositivo. Se você vir esse status exibido, isso significa que o tipo de algoritmo de fragmentação usado não pode ser aplicado à atualização de firmware do seu dispositivo. O erro ocorre porque o dispositivo não tem o pacote FUOTA suportado. Para resolver esse erro, verifique se o firmware do dispositivo pode receber atualizações de firmware usando o FUOTA.

- Não há memória suficiente

  DepoisAWS IoT Core for LoRaWANenvia os fragmentos de imagem, seus dispositivos finais coletam os fragmentos de imagem e reconstroem a imagem binária desses fragmentos. Esse status é exibido quando o dispositivo não tem memória suficiente para montar os fragmentos de entrada da imagem do firmware, o que pode resultar no término prematuro da sessão de atualização do firmware. Para resolver o erro, verifique se o hardware do dispositivo pode receber essa atualização. Se o dispositivo não puder receber essa atualização, use uma imagem delta para atualizar o firmware.

- Índice de fragmentação sem suporte

  O índice de fragmentação identifica uma das quatro sessões de fragmentação possíveis simultaneamente. Se o dispositivo não suportar o valor do índice de fragmentação indicado, esse status será exibido. Para resolver esse erro, execute um ou mais dos procedimentos a seguir.

  - Inicie uma nova tarefa FUOTA para o dispositivo.
  - Se o erro persistir, alterne do unicast para o modo multicast.
  - Se o erro ainda não for resolvido, verifique o firmware do dispositivo.

- Erro de memória

  Esse status indica que o dispositivo sofreu um erro de memória ao receber os fragmentos de entrada deAWS IoT Core for LoRaWAN. Se esse erro ocorrer, talvez seu dispositivo não consiga receber essa atualização. Para resolver o erro, verifique se o hardware do dispositivo pode receber essa atualização. Se necessário, use uma imagem delta para atualizar o firmware do dispositivo.

- Descritor errado

  Seu dispositivo não oferece suporte a o descritor indicado. O descritor é um campo que descreve o arquivo que será transportado durante a sessão de fragmentação. Se você vir esse erro, entre em contato comAWS SupportCentro.

- Replay de contagem de sessões

  Esse status indica que o dispositivo já usou essa contagem de sessões anteriormente. Para resolver o erro, inicie uma nova tarefa FUOTA para o dispositivo.

- Fragmentos em falta

  À medida que seu dispositivo coleta os fragmentos de imagem deAWS IoT Core for LoRaWAN, ele reconstrói a nova imagem de firmware a partir dos fragmentos codificados independentes. Se o dispositivo não recebeu todos os fragmentos, a nova imagem não poderá ser reconstruída e você verá esse status. Para resolver o erro, inicie uma nova tarefa FUOTA para o dispositivo.

- Erro de MIC

  Quando o dispositivo reconstrói a nova imagem de firmware dos fragmentos coletados, ele executa um MIC (Verificação de Integridade da Mensagem) para verificar a autenticidade da imagem e se ela vem da fonte correta. Se o dispositivo detectar uma incompatibilidade no MIC após remontar os fragmentos, esse status será exibido. Para resolver o erro, inicie uma nova tarefa FUOTA para o dispositivo.
- Bem-sucedida

  A sessão FUOTA para seu dispositivo foi bem-sucedida.

  > **Note**
  >
  > Embora essa mensagem de status indique que os dispositivos reconstruíram a imagem a partir dos fragmentos e a verificaram, o firmware do dispositivo pode não ter sido atualizado quando o dispositivo relatar o status paraAWS IoT Core for LoRaWAN. Verifique se o firmware do dispositivo foi atualizado.

Next steps (Próximas etapas)

Você aprendeu sobre os diferentes status da tarefa FUOTA e seus dispositivos e como solucionar problemas. Para obter mais informações sobre cada um desses status, consulte aEspecificação de transporte de bloco de dados fragmentados LoRaWAN, TS004-1.0.0.

Monitorando sua frota de recursos sem fio em tempo real usando o analisador de rede

O analisador de rede usa um padrão WebSocket conexão para receber registros de mensagens de rastreamento em tempo real para seus recursos de conectividade sem fio. Usando o analisador de rede, você pode adicionar os recursos que deseja monitorar, ativar uma sessão de mensagens de rastreamento e começar a receber mensagens de rastreamento em tempo real.

Para monitorar seus recursos, você também pode usar o Amazon CloudWatch. Para usar o CloudWatch, você configura uma função do IAM para configurar o log e aguarda que as entradas de registro sejam exibidas no console. O analisador de rede reduz significativamente o tempo necessário para configurar uma conexão e começar a receber mensagens de rastreamento, fornecendo a você just-in-time informações de registro para sua frota de recursos. Para obter informações sobre o monitoramento usando o CloudWatch, consulteMonitorar e registrar em log paraAWS IoT WirelessUsar o Amazon CloudWatch (p. 1226).

Ao reduzir o tempo de configuração e usar as informações das mensagens de rastreamento, você pode monitorar seus recursos com mais eficiência, obter insights significativos e solucionar erros. Você pode monitorar os dispositivos LoraWAN e os gateways LoraWAN. Por exemplo, você pode identificar rapidamente um erro de junção ao integrar um de seus dispositivos LoraWAN. Para depurar o erro, use as informações no log de mensagens de rastreamento fornecido.

Como usar o analisador de rede

Para monitorar sua frota de recursos e começar a receber mensagens de rastreamento, execute as seguintes etapas

1. Crie a configuração do analisador de rede e adicione recursos

   Antes de ativar as mensagens de rastreamento, crie uma configuração do analisador de rede e adicione recursos à sua configuração. Primeiro, especifique as configurações, que incluem níveis de log e informações de quadro do dispositivo sem fio. Em seguida, adicione os recursos sem fio que você deseja monitorar usando o gateway sem fio e os identificadores de dispositivos sem fio.

2. Transmita mensagens de rastreamento com WebSockets

   Você pode gerar uma URL de solicitação pré-assinada usando as credenciais da sua função do IAM para transmitir mensagens de rastreamento do analisador de rede usando o WebSocket Protocolo.
3. Ativar sessão de mensagens de rastreamento e monitorar mensagens de rastreamento

   Para começar a receber mensagens de rastreamento, ative sua sessão de mensagens de rastreamento. Para evitar incorrer em custos adicionais, você pode desativar ou fechar a sessão de mensagens de rastreamento do analisador de rede.

A seguir, mostra como criar sua configuração, adicionar recursos e ativar sua sessão de mensagens de rastreamento.

Tópicos

- Adicione a função do IAM necessária para o analisador de rede (p. 1202)
- Crie uma configuração de analisador de rede e adicione recursos (p. 1204)
- Transmita mensagens de rastreamento do analisador de rede com WebSockets (p. 1209)
- Visualize e monitore registros de mensagens de rastreamento do analisador de rede em tempo real (p. 1215)

Adicione a função do IAM necessária para o analisador de rede

Ao usar o analisador de rede, você deve conceder a um usuário permissão para usar as operações da APIUpdateNetworkAnalyzerConfigurationeGetNetworkAnalyzerConfigurationpara acessar recursos do analisador de rede. A seguir, mostra as políticas do IAM que você usa para conceder permissões.

Políticas do IAM para analisador de rede

Use um dos seguintes:

- Política sem fio de acesso total

  GrantAWS IoT Corepara a LoraWan a política de acesso total anexando a políticaAWSIoTWirelessFullAccessPara sua função. Para obter mais informações, consulte`AWSIoTWirelessFullAccess`Resumo da política.
- Política do IAM com escopo para a API Get and Update

  Crie a seguinte política do IAM acessando oCriar políticapágina do console do IAM e naEditor visualGuia:
  1. SelecioneIoTWirelesspeloServiço.
  2. UnderNível de acesso, expandaLeiae escolhaGetNetworkAnalyzerConfiguratione, em seguida, expandaGravaçãoe escolhaUpdateNetworkAnalyzerConfiguration.
  3. SelecionePróximo: tagse insira umName (Nome)para a política, comoIOTWirelessNetworkAnalyzerPolicy. Escolha Create policy (Criar política).

  A seguir, mostra a políticaIOTWirelessNetworkAnalyzerPolicyque você criou. Para obter mais informações sobre como criar uma política, consulteCrie políticas do IAM.

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

```
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": [
                "iotwireless:GetNetworkAnalyzerConfiguration",
                "iotwireless:UpdateNetworkAnalyzerConfiguration"
            ],
            "Resource": "*"
        }
    ]
}
```

Política com escopo para acessar recursos específicos

Para configurar um controle de acesso mais fino, você deve adicionar os gateways e dispositivos sem fio àRecursocampo. A política a seguir usa o ARN curinga para conceder acesso a todos os gateways e dispositivos. Você pode controlar o acesso a gateways e dispositivos específicos usando o`WirelessGatewayIdeWirelessDeviceId`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": [
                "iotwireless:GetNetworkAnalyzerConfiguration",
                "iotwireless:UpdateNetworkAnalyzerConfiguration"
            ],
            "Resource": [
                "arn:aws:iotwireless:*:{accountId}:WirelessDevice/*",
                "arn:aws:iotwireless:*:{accountId}:WirelessGateway/*",
                "arn:aws:iotwireless:*:{accountId}:NetworkAnalyzerConfiguration/*"
            ]
        }
    ]
}
```

Para conceder a um usuário permissão para usar o analisador de rede, mas não usar nenhum gateway ou dispositivo sem fio, use a política a seguir. A menos que especificado, as permissões para usar os recursos são implicitamente negadas.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": [
                "iotwireless:GetNetworkAnalyzerConfiguration",
                "iotwireless:UpdateNetworkAnalyzerConfiguration"
            ],
            "Resource": [
                "arn:aws:iotwireless:*:{accountId}:NetworkAnalyzerConfiguration/*"
            ]
        }
    ]
}
```

Next steps (Próximas etapas)

Agora que você criou a política, você pode adicionar recursos à configuração do analisador de rede e receber informações de mensagens de rastreamento desses recursos. Para obter mais informações, consulteCrie uma configuração de analisador de rede e adicione recursos (p. 1204)

Crie uma configuração de analisador de rede e adicione recursos

Antes de poder transmitir mensagens de rastreamento, crie uma configuração de analisador de rede e adicione os recursos que você deseja monitorar a essa configuração. Ao criar uma configuração, é possível:

- Especifique um nome de configuração e uma descrição opcional.
- Personalize as configurações, como informações do quadro e nível de detalhes para suas mensagens de registro.
- Adicione os recursos que você deseja monitorar. Os recursos podem ser dispositivos sem fio ou gateways sem fio, ou ambos.

As configurações especificadas determinarão as informações de mensagens de rastreamento que você receberá para os recursos adicionados à configuração. Você também pode querer criar várias configurações, dependendo do seu caso de uso de monitoramento.

A tabela a seguir mostra como criar uma configuração e adicionar recursos.

Tópicos

- Criar uma configuração de analisador de rede (p. 1204)
- Adicione recursos e atualize a configuração do analisador de rede (p. 1207)

Criar uma configuração de analisador de rede

Antes de monitorar seus gateways sem fio ou dispositivos sem fio, você deve criar uma configuração de analisador de rede. Ao criar a configuração, você só precisa especificar um nome de configuração. Você pode personalizar suas configurações e adicionar os recursos que deseja monitorar à configuração mesmo depois que ela for criada. As configurações determinam as informações de mensagens de rastreamento que você receberá para esses recursos.

Dependendo dos recursos que você deseja monitorar e do nível de informações que deseja receber para eles, talvez você queira criar várias configurações. Por exemplo, você pode criar uma configuração que exibe somente informações de erro para um conjunto de gateways em seuConta da AWS. Você também pode criar uma configuração que exiba todas as informações sobre um dispositivo sem fio que você deseja monitorar.

As seções a seguir mostram as várias definições de configuração e como criar sua configuração.

Definições de configuração

Ao criar ou atualizar a configuração do analisador de rede, você também pode personalizar os seguintes parâmetros para filtrar as informações do fluxo de log.

- Informações sobre o quadro

  Essa configuração é a informação do quadro para os recursos do dispositivo sem fio para mensagens de rastreamento. As informações do quadro podem ser usadas para depurar a comunicação entre o servidor de rede e os dispositivos finais. Ele é habilitado por padrão.

- Níveis de log

  Você pode visualizar registros de informações ou erros ou desativar o registro em log.

  - Informações

    Logs com um nível de log deInformaçõessão mais detalhados e contêm fluxos de log de erros e fluxos de log informativos. Os logs informativos podem ser usados para exibir alterações no estado de um dispositivo ou gateway.

    > **Note**
    >
    > Coletar fluxos de log mais detalhados pode incorrer em custos adicionais. Para obter mais informações sobre a definição de preço, consulte Definição de preço do AWS IoT Core.

  - Erro

    Logs com um nível de log deErrosão menos detalhados e exibem apenas informações de erro. Você pode usar esses registros quando um aplicativo tiver um erro, como um erro de conexão de dispositivo. Usando as informações do fluxo de log, você pode identificar e solucionar problemas de recursos em sua frota.

Criar uma configuração usando o console

Você pode criar uma configuração do analisador de rede e personalizar os parâmetros opcionais usando oAWS IoTconsole ou oAWS IoTAPI Wireless. Você também pode criar várias configurações e, posteriormente, excluir todas as configurações que não estiver mais sendo usada.

Criar uma configuração de analisador de rede

1. Abrir o Hub Network Analyzer doAWS IoTconsolee escolhaCriar configurações.
2. Especifique as definições de configuração.
   - Nome, descrição e etiquetas

     Especifique um exclusivoNome da configuraçãoqueSó contém letras, números, hifens ou sublinhados. Use o opcionalDescriçãocampo para fornecer informações sobre a configuração e oTagscampo para adicionar pares de metadados de chave-valor sobre a configuração. Para obter mais informações sobre como nomear e descrever seus recursos, consulteDescreva suasAWS IoT Corepara recursos LoRaWAN (p. 1131).

   - Definições de configuração

     Escolha se deseja desativar as informações do quadro e usarSelecione níveis de logPara escolher os níveis de log que você deseja usar para seus logs de mensagens de rastreamento. Escolha Next (Próximo).
3. Adicione recursos à configuração. Você pode adicionar seus recursos agora ou escolherCriare, em seguida, adicione seus recursos mais tarde. Para adicionar recursos mais tarde, escolhaCriar.

   NoPágina hub do Network Analyzer, você verá a configuração que criou junto com suas configurações. Para visualizar os detalhes da nova configuração, selecione o nome da configuração.

Exclua a configuração do analisador de rede

Você pode criar várias configurações do analisador de rede dependendo dos recursos que deseja monitorar e do nível de informações de mensagens de rastreamento que deseja receber para eles.

Para remover configurações do console

1. Vá para oHub Network Analyzer doAWS IoTconsolee escolha a configuração que você deseja remover.
2. Escolha Actions (Ações) e, em seguida, escolha Delete (Excluir).

Criar uma configuração usando a API

Para criar uma configuração de analisador de rede usando a API, use oCreateNetworkAnalyzerConfigurationOperação da API docreate-network-analyzer-configurationComando da CLI.

Ao criar sua configuração, você só precisa especificar um nome de configuração. Você também pode usar essa operação de API para especificar as configurações e adicionar recursos ao criar a configuração. Como alternativa, você pode especificá-los mais tarde usando oUpdateNetworkAnalyzerConfigurationOperação da API doupdate-network-analyzer-configurationCLI.

- Criar uma configuração

  Ao criar sua configuração, você deve especificar um nome. Por exemplo, o seguinte comando cria uma configuração fornecendo apenas um nome e uma descrição opcional. Por padrão, a configuração tem informações de quadro ativadas e usa um nível de log de`INFO`.

```
aws iotwireless create-network-analyzer-configuration \
    --configuration-name My_Network_Analyzer_Config \
    --description "My first network analyzer configuration"
```

  A execução deste comando exibe o ARN e o ID da configuração do analisador de rede.

```
{
    "Arn": "arn:aws:iotwireless:us-
east-1:123456789012:NetworkAnalyzerConfiguration/12345678-a1b2-3c45-67d8-e90fa1b2c34d",
    "Id": "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
}
```

- Criar configuração com recursos

  Para personalizar as definições de configuração, use o`trace-content`parâmetro . Para adicionar recursos, use o`WirelessDevices`e`WirelessGateways`Para especificar os gateways, dispositivos ou ambos que você deseja adicionar à sua configuração. Por exemplo, o comando a seguir personaliza as configurações e adiciona à sua configuração os recursos sem fio, especificados por seus`WirelessGatewayID`e`WirelessDeviceID`.

```
aws iotwireless create-network-analyzer-configuration \
    --configuration-name My_NetworkAnalyzer_Config \
    --trace-content WirelessDeviceFrameInfo=DISABLED,LogLevel="ERROR" \
    --wireless-gateways "12345678-a1b2-3c45-67d8-e90fa1b2c34d" "90123456-de1f-2b3b-4c5c-
bb1112223cd1"
    --wireless-devices "1ffd32c8-8130-4194-96df-622f072a315f"
```

  O exemplo a seguir mostra a saída da execução do comando:

```
{
    "Arn": "arn:aws:iotwireless:us-
east-1:123456789012:NetworkAnalyzerConfiguration/12345678-a1b2-3c45-67d8-e90fa1b2c34d",
    "Id": "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
}
```

Listar configurações do analisador de rede

Você pode criar várias configurações do analisador de rede dependendo dos recursos que deseja monitorar e do nível de detalhes das informações de mensagens de rastreamento que deseja receber para os recursos. Depois de criar essas configurações, você pode usar oListNetworkAnalyzerConfigurationsOperação da API dolist-network-analyzer-configurationComando da CLI para obter uma lista dessas configurações.

```
aws iotwireless list-network-analyzer-configurations
```

A execução deste comando exibe todas as configurações do analisador de rede em seuConta da AWS. Você também pode usar `omax-results`parâmetro para especificar quantas configurações você deseja exibir. A tabela a seguir mostra a saída da execução desse comando.

```
{
   "NetworkAnalyzerConfigurationList": [
      {
         "Arn": "arn:aws:iotwireless:us-
east-1:123456789012:NetworkAnalyzerConfiguration/12345678-a1b2-3c45-67d8-e90fa1b2c34d",
         "Name": "My_Network_Analyzer_Config1"
      },
      {
         "Arn": "arn:aws:iotwireless:us-
east-1:123456789012:NetworkAnalyzerConfiguration/90123456-a1a2-9a87-65b4-c12bf3c2d09a",
         "Name": "My_Network_Analyzer_Config2"
      }
   ]
}
```

Exclua a configuração do analisador de rede

Você pode excluir uma configuração que não está mais sendo usada com oDeleteNetworkAnalyzerConfigurationOperação da API dodelete-network-analyzer-configurationComando da CLI.

```
aws iotwireless delete-network-analyzer-configuration \
    --configuration-name My_NetworkAnalyzer_Config
```

A execução desse comando não produz nenhuma saída. Para ver as configurações disponíveis, você pode usar o`ListNetworkAnalyzerConfigurations`Operação da API do.

Next steps (Próximas etapas)

Agora que você criou uma configuração do analisador de rede, você pode adicionar recursos à sua configuração ou atualizar suas configurações. Para obter mais informações, consulteAdicione recursos e atualize a configuração do analisador de rede (p. 1207)

Adicione recursos e atualize a configuração do analisador de rede

Antes de ativar as mensagens de rastreamento, adicione os recursos que você deseja monitorar à configuração do analisador de rede. Os recursos podem ser um ou ambos os dispositivos LoraWAN e gateways LoraWAN.

Pré-requisitos

Antes que você possa adicionar recursos:

1. Você deve ter integrado os gateways e dispositivos que deseja monitorarAWS IoT Corepara LoRaWAN. Para obter mais informações, consulteConectando gateways e dispositivos aAWS IoT Core for LoRaWAN (p. 1130)
2. Você deve ter criado uma configuração do analisador de rede para a qual você adicionará recursos. Para obter mais informações, consulteCriar uma configuração de analisador de rede (p. 1204)

Adicione recursos e atualize configurações usando o console

Você pode adicionar recursos e personalizar os parâmetros opcionais usando oAWS IoTconsole ou oAWS IoTAPI Wireless. Além dos recursos, você também pode editar suas configurações e salvar a configuração atualizada.

Adicione recursos à configuração

1. Abrir oHub Network Analyzer doAWS IoTconsoleE escolha a configuração para a qual você deseja adicionar recursos.
2. SelecioneAçõese, depois, escolhaAdicione recursos.
3. Adicione os recursos que você deseja monitorar usando o gateway sem fio e os identificadores de dispositivos sem fio. Você pode escolher vários recursos e adicionar até 250 gateways sem fio ou dispositivos sem fio.
4. Depois de adicionar todos os recursos, escolhaAdicionar.

   Você verá o número de gateways e dispositivos que você adicionou naPágina hub do Network Analyzer. Você pode continuar adicionando e removendo recursos até ativar a sessão de mensagens de rastreamento. Depois que a sessão for ativada, para adicionar recursos, você terá que desativar a sessão.

Atualizar as definições de configuração

1. Abrir oHub Network Analyzer doAWS IoTconsoleE escolha a configuração para a qual deseja atualizar as configurações.
2. Escolha Actions (Ações) e escolha Edit (Editar).
3. Escolha se deseja desativar as informações do quadro e usarSelecione níveis de logPara escolher os níveis de log que você deseja usar para seus logs de mensagens de rastreamento. Escolha Save (Salvar).

   Você verá as configurações especificadas na página de detalhes da configuração do analisador de rede.

Adicione recursos e atualize as configurações usando a API

Para adicionar recursos ou atualizar suas configurações, use oUpdateNetworkAnalyzerConfigurationAPI ou o comandoupdate-network-analyzer-configurationCLI.

- Atualizar configurações

  Para atualizar as definições de configuração, use o`TraceContent`parâmetro para especificar o nível de log e se deseja ativar as informações do quadro. Por exemplo, o comando a seguir atualiza as configurações desativando as informações do quadro e definindo o nível de log como`ERROR`.

```
aws iotwireless update-network-analyzer-configuration \
    --configuration-name NetworkAnalyzerConfig_Default \
    --trace-content WirelessDeviceFrameInfo=DISABLED,LogLevel="ERROR"
```

- Adicione recursos

Para adicionar recursos, use o`WirelessDevicesToAdd`e`WirelessGatewaysToAdd`Para especificar os gateways ou dispositivos ou ambos que você deseja adicionar à sua configuração. Por exemplo, o comando a seguir atualiza as configurações e adiciona à sua configuração os recursos sem fio, especificados por seus`WirelessGatewayID`e`WirelessDeviceID`.

```
aws iotwireless update-network-analyzer-configuration \
    --configuration-name NetworkAnalyzerConfig_Default \
    --trace-content WirelessDeviceFrameInfo=DISABLED,LogLevel="ERROR" \
    --wireless-gateways-to-add "12345678-a1b2-3c45-67d8-e90fa1b2c34d" "90123456-
de1f-2b3b-4c5c-bb1112223cd1"
    --wireless-devices-to-add "1ffd32c8-8130-4194-96df-622f072a315f"
```

Para remover dispositivos ou gateways, use o`WirelessDevicesToRemove`e`WirelessGatewaysToRemove`parâmetros da API.

Obtenha informações sobre a configuração

Executar o`UpdateNetworkAnalyzerConfiguration`A API não produz nenhuma saída. Para visualizar suas configurações e os gateways ou dispositivos que você adicionou, use oGetNetworkAnalyzerConfigurationOperação da API doget-network-analyzer-configurationcomando. Forneça o nome da configuração do analisador de rede como entrada.

```
aws iotwireless get-network-analyzer-configuration \
    --configuration-name NetworkAnalyzerConfig_Default
```

A execução desse comando produz a seguinte saída.

```
{
    "TraceContent": {
        "WirelessDeviceFrameInfo": "DISABLED",
        "LogLevel": "ERROR"
    },
    "WirelessDevices": [],
    "WirelessGateways": [
        "a0dd70e5-8f15-41a5-89cf-310284e691a1",
        "41682155-de4f-4f8f-84bf-bb5557221fc8"
    ]
}
```

Next steps (Próximas etapas)

Agora que você adicionou recursos e especificou todas as configurações opcionais para sua configuração, você pode usar o WebSocket Protocolo para estabelecer uma conexão comAWS IoT Corepara LoRaWAN por usar o analisador de rede. Em seguida, você pode ativar as mensagens de rastreamento e começar a receber mensagens de rastreamento para seus recursos. Para obter mais informações, consulteTransmita mensagens de rastreamento do analisador de rede com WebSockets (p. 1209)

Transmita mensagens de rastreamento do analisador de rede com WebSockets

Quando você usa o WebSocket , você pode transmitir mensagens de rastreamento do analisador de rede em tempo real. Quando você envia uma solicitação, o serviço responde com uma estrutura JSON. Depois de ativar as mensagens de rastreamento, você pode usar os registros de mensagens para obter informações sobre seus recursos e solucionar erros. Para obter mais informações, consulteProtocolo WebSocket.

A seguir, mostra como transmitir mensagens de rastreamento do analisador de rede com WebSockets.

Tópicos

- Gerar uma solicitação pré-assinada com a WebSocket biblioteca (p. 1210)
- Mensagens e códigos de status WebSocket (p. 1213)

Gerar uma solicitação pré-assinada com a WebSocket biblioteca

A seguir mostra como você gerar uma solicitação pré-assinada para que você possa usar o WebSocket biblioteca para enviar solicitações para o serviço.

Adicione uma política para WebSocket Solicitações à função do IAM

Para usar a WebSocket Protocolo para chamar o analisador de rede, anexar a seguinte política aoAWS Identity and Access Management(IAM) função que faz essa solicitação.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "iotwireless:StartNetworkAnalyzerStream",
            "Resource": "*"
        }
    ]
}
```

Criar um URL pré-assinado

Construa um URL para seu WebSocket Solicitação que contém as informações necessárias para configurar a comunicação entre o aplicativo e o analisador de rede. Para verificar a identidade da solicitação, WebSocket O streaming usa o processo do Amazon Signature versão 4 para assinar solicitações. Para obter mais informações sobre Signature versão 4, consulteAssinaturaAWSSolicitações da API donoReferência geral do Amazon Web Services.

Para chamar o analisador de rede, use o`StartNetworkAnalyzerStream`URL da solicitação. A solicitação será assinada usando as credenciais para a função do IAM mencionada anteriormente. O URL tem o seguinte formato com quebras de linha adicionadas para legibilidade. Você deve adicionar o nome da configuração abaixo da linha`&X-Amz-SignedHeaders=host`. Quaisquer parâmetros adicionais devem ser adicionados abaixo desta linha classificada por ordem alfabética.

O exemplo a seguir mostra como usar esse URL de solicitação com o nome da configuração,*`NaConfig`*:

```
wss://api.iotwireless.<region>.amazonaws.com/start-network-analyzer-stream?configuration-
name=NaConfig
    &X-Amz-Algorithm=AWS4-HMAC-SHA256
    &X-Amz-Date=20220427T001057Z&X-Amz-SignedHeaders=host
    &X-Amz-Expires=300
    &X-Amz-Credential=credential_number/account/region/iotwireless/aws4_request
    &X-Amz-Signature=c123456789098765a012c3a45d6789dd01234af5678bba9bbc0dbc112a3334d
```

> **Note**
>
> Se o URL não incluir o nome da configuração,AWS IoT Corepara LoRaWAN incluirá o nome padrão para a configuração do analisador de rede,`NetworkAnalyzerConfig_Default`.

```
GET wss://api.iotwireless.<region>.amazonaws.com/start-network-analyzer-stream?X-Amz-
Algorithm=AWS4-HMAC-SHA256
```

```
&X-Amz-Credential=Signature Version 4 credential scope
&X-Amz-Date=date
&X-Amz-Expires=time in seconds until expiration
&X-Amz-Security-Token=security-token
&X-Amz-Signature=Signature Version 4 signature
&X-Amz-SignedHeaders=host
```

Use os seguintes valores para os parâmetros do Signature versão 4:

- X-Amz-Algoritmo— O algoritmo que você está usando no processo de assinatura. O único valor válido é `AWS4-HMAC-SHA256`.
- X-Amz-Credential— Uma string separada por barras ("/"), formada concatenando seu ID de chave de acesso e seus componentes de escopo de credencial. O escopo da credencial inclui a data no formato AAAAMMDD, oAWSRegião, o nome do serviço e uma string de encerramento (aws4_request).
- X-Amz-Date— A data e hora em que a assinatura foi criada. Gere a data e a hora seguindo as instruções noTratamento de datas no Signature versão 4noReferência geral do Amazon Web Services.
- X-Amz-Expira— O tempo, em segundos, até que as credenciais expirem. O valor máximo é 300 segundos (5 minutos).
- X-Amz-Security-Token— (opcional) Um token Signature versão 4 para credenciais temporárias. Se você especificar esse parâmetro, inclua-o na solicitação canônica. Para obter mais informações, consulteSolicitação de credenciais de segurança temporáriasnoAWSGuia do usuário Identity and Access Management.
- X-Amz-Signature— A assinatura do Signature versão 4 que você gerou para a solicitação.
- X-Amz-SignedHeaders— Os cabeçalhos que são assinados ao criar a assinatura para a solicitação. O único valor válido é `host`.

Crie o URL da solicitação e crie a assinatura do Signature versão 4

Para criar o URL para a solicitação e criar a assinatura do Signature versão 4, siga as etapas a seguir. Os exemplos estão em pseudocódigo.

Tarefa 1: Criar uma solicitação canônica

Crie uma string que inclua as informações da solicitação em um formato padronizado. Isso garante que quandoAWSRecebe a solicitação, ela pode calcular a mesma assinatura que você calcular noTarefa 3: Calcular a assinatura (p. 1213). Para obter mais informações, consulteCriar uma solicitação canônica para Signature versão 4noReferência geral do Amazon Web Services.

1. Defina variáveis para a solicitação no seu aplicativo.

```
# HTTP verb
method = "GET"
# Service name
service = "iotwireless"
# Região da AWS
region = "Região da AWS"
# Service streaming endpoint
endpoint = "wss://api.iotwireless.region.amazonaws.com"
# Host
host = "api.iotwireless.<region>.amazonaws.com"
# Date and time of request
amz-date = YYYYMMDD'T'HHMMSS'Z'
# Date without time for credential scope
datestamp = YYYYMMDD
```

2. Crie um URI canônico (identificador de recurso uniforme). O URI canônico é a parte do URI entre o domínio e a string de consulta.

```
canonical_uri = "/start-network-analyzer-stream"
```

3. Crie cabeçalhos canônicos e cabeçalhos assinados. Observe o `\n` que vem no final dos cabeçalhos canônicos.

   - Acrescente o nome do cabeçalho em minúscula seguido por ponto e vírgula.
   - Anexe uma lista separada por vírgulas de valores para esse cabeçalho. Não classifique os valores em cabeçalhos que têm vários valores.
   - Anexe uma nova linha (`\n`).

```
canonical_headers = "host:" + host + "\n"
signed_headers = "host"
```

4. Faça uma correspondência entre o algoritmo e o algoritmo de hash. Você deve usar SHA-256.

```
algorithm = "AWS4-HMAC-SHA256"
```

5. Crie o escopo da credencial, que define o escopo da chave derivada como a data, a Região e o serviço ao qual a solicitação foi feita.

```
credential_scope = datestamp + "/" + region + "/" + service + "/" + "aws4_request"
```

6. Crie a string de consulta canônica. Os valores de string de consulta devem ser codificados em URI e classificados por nome.

   - Classifique os nomes de parâmetro por ponto de código de caracteres em ordem ascendente. Parâmetros com nomes duplicados devem ser classificados pelo valor. Por exemplo, um nome de parâmetro que começa com a letra maiúscula F precede um nome de parâmetro que começa com uma letra minúscula b.
   - Não codificar em URI nenhum dos caracteres não reservados queRFC 3986define: A—Z, a—z, 0—9, hífen (-), sublinhado (_), ponto (.) e til (~).
   - Codificar em percentual todos os outros caracteres com %XY, onde X e Y são caracteres hexadecimais (de 0 a 9 e maiúsculas de A a F). Por exemplo, o caractere de espaço deve ser codificado como%20 (não usando '+', como alguns esquemas de codificação), e deve haver caracteres UTF-8 estendidos no formulário%XY%ZA%BC.
   - Codifique duas vezes todos os caracteres de sinal de igual (=) em valores de parâmetro.

```
canonical_querystring  = "X-Amz-Algorithm=" + algorithm
canonical_querystring += "&X-Amz-Credential="+ URI-encode(access key + "/" +
 credential_scope)
canonical_querystring += "&X-Amz-Date=" + amz_date
canonical_querystring += "&X-Amz-Expires=300"
canonical_querystring += "&X-Amz-Security-Token=" + token
canonical_querystring += "&X-Amz-SignedHeaders=" + signed_headers
```

7. Crie um hash da carga. Para uma solicitação GET, a carga é uma string vazia.

```
payload_hash = HashSHA256(("").Encode("utf-8")).HexDigest()
```

8. Por fim, combine todos os elementos para criar a solicitação canônica.

```
canonical_request = method + '\n'
   + canonical_uri + '\n'
   + canonical_querystring + '\n'
```

```
    + canonical_headers + '\n'
    + signed_headers + '\n'
    + payload_hash
```

Tarefa 2: Criar a string para assinar

A string para assinar inclui metainformações sobre sua solicitação. Você usará a string para assinar na próxima etapa, quando calcular a assinatura da solicitação. Para obter mais informações, consulteCriar uma string para assinar a Signature versão 4noReferência geral do Amazon Web Services.

```
string_to_sign=algorithm + "\n"
   + amz_date + "\n"
   + credential_scope + "\n"
   + HashSHA256(canonical_request.Encode("utf-8")).HexDigest()
```

Tarefa 3: Calcular a assinatura

Você derivará uma chave de assinatura da sua chave de acesso secreta da AWS. Para um grau maior de proteção, a chave derivada é específica para a data, o serviço eAWSRegião : Você usa a chave derivada para assinar a solicitação. Para obter mais informações, consulteCalcular a assinatura paraAWSSignature versão 4noReferência geral do Amazon Web Services.

O código pressupõe que você tenha implementado a função `GetSignatureKey` para gerar uma chave de assinatura. Para obter mais informações e funções de exemplo, consulteExemplos de Como obter uma chave de assinatura do Signature versão 4noReferência geral do Amazon Web Services.

A função `HMAC(key, data)` representa uma função HMAC-SHA256 que retorna os resultados em formato binário.

```
#Create the signing key
signing_key = GetSignatureKey(secret_key, datestamp, region, service)

# Sign the string_to_sign using the signing key
signature = HMAC.new(signing_key, (string_to_sign).Encode("utf-8"), Sha256()).HexDigest
```

Tarefa 4: Adicionar informações de assinatura para solicitar e criar URL de solicitação

Depois de calcular a assinatura, adicione-a à string de consulta. Para obter mais informações, consulteAdicione a assinatura à solicitaçãonoReferência geral do Amazon Web Services.

```
#Add the authentication information to the query string
canonical_querystring += "&X-Amz-Signature=" + signature

# Sign the string_to_sign using the signing key
request_url = endpoint + canonical_uri + "?" + canonical_querystring
```

Next steps (Próximas etapas)

Agora, você pode usar o URL da solicitação com o WebSocket Biblioteca para fazer a solicitação ao serviço e observar as mensagens. Para obter mais informações, consulteMensagens e códigos de status WebSocket (p. 1213)

Mensagens e códigos de status WebSocket

Depois de criar uma solicitação pré-assinada, você pode usar o URL da solicitação com seu WebSocket biblioteca, ou uma biblioteca adequada à sua linguagem de programação, para fazer solicitações ao

serviço. Para obter mais informações sobre como você pode gerar essa solicitação pré-assinada, consulteGerar uma solicitação pré-assinada com a WebSocket biblioteca (p. 1210).

Mensagens WebSocket

O WebSocket protocolo pode ser usado para estabelecer uma conexão bidirecional. As mensagens podem ser transmitidas de cliente para servidor e de servidor para cliente. No entanto, o analisador de rede suporta somente mensagens enviadas de servidor para cliente. Qualquer mensagem recebida do cliente é inesperada e o servidor fechará automaticamente o WebSocket conexão se uma mensagem for recebida do cliente.

Quando a solicitação é recebida e uma sessão de mensagens de rastreamento é iniciada, o servidor responde com uma estrutura JSON, que é a carga útil. Para obter mais informações sobre a carga útil e como você pode ativar as mensagens de rastreamento a partir doAWS Management Console, consulteVisualize e monitore registros de mensagens de rastreamento do analisador de rede em tempo real (p. 1215).

Códigos de status do WebSocket

Veja a seguir a WebSocket códigos de status para a comunicação do servidor para o cliente. O WebSocket códigos de status seguem aPadrão RFC de fechamento normal de conexões.

A seguir, mostra os códigos de status suportados:

- 1000

  Esse código de status indica um fechamento normal, o que significa que o WebSocket a conexão foi estabelecida e a solicitação foi atendida. Esse status pode ser observado quando uma sessão está ociosa, fazendo com que a conexão atinja o tempo limite.
- 1.002

  Esse código de status indica que o endpoint está encerrando a conexão devido a um erro de protocolo.
- 1003

  Esse código de status indica um status de erro em que o endpoint encerrou a conexão porque recebeu dados em um formato que ele não pode aceitar. O endpoint suporta somente dados de texto e pode exibir esse código de status se receber uma mensagem binária ou uma mensagem do cliente que esteja usando um formato não suportado.
- 1008

  Esse código de status indica um status de erro em que o endpoint encerrou a conexão porque recebeu uma mensagem que viola sua política. Esse status é genérico e é exibido quando os outros códigos de status, como 1003 ou 1009, não são aplicáveis. Você também verá esse status exibido se houver necessidade de ocultar a política ou quando houver uma falha de autorização, como uma assinatura expirada.
- 1011

  Esse código de status indica um status de erro em que o servidor está encerrando a conexão porque encontrou uma condição inesperada ou um erro interno que o impediu de atender a solicitação.

Next steps (Próximas etapas)

Agora que você aprendeu a gerar uma solicitação pré-assinada e como observar mensagens do servidor usando o WebSocket conexão, você pode ativar as mensagens de rastreamento e começar a receber registros de mensagens para o gateway sem fio e os recursos do dispositivo sem fio. Para obter mais informações, consulteVisualize e monitore registros de mensagens de rastreamento do analisador de rede em tempo real (p. 1215)

Visualize e monitore registros de mensagens de rastreamento do analisador de rede em tempo real

Se você adicionou recursos à configuração do analisador de rede, poderá ativar as mensagens de rastreamento para começar a receber mensagens de rastreamento para seus recursos. É possível usar oAWS Management Console, oAWS IoTAPI sem fio ou oAWS CLI.

Pré-requisitos

Antes de poder ativar as mensagens de rastreamento usando o analisador de rede, você deve ter:

- Adicionados os recursos que você deseja que o monitore à configuração padrão do analisador de rede. Para obter mais informações, consulteAdicione recursos e atualize a configuração do analisador de rede (p. 1207)
- Gerou uma solicitação pré-assinada usando o`StartNetworkAnalyzerStream`URL da solicitação. A solicitação será assinada usando as credenciais para oAWS Identity and Access Managementfunção que faz essa solicitação. Para obter mais informações, consulteCriar um URL pré-assinado (p. 1210)

Ativar mensagens de rastreamento usando o console

Para ativar o rastreamento de mensagens

1. Abrir oHub Network Analyzer doAWS IoTconsolee escolha a configuração do analisador de rede,NetworkAnalyzerConfig_default.
2. Na página de detalhes da configuração do analisador de rede, escolhaAtivar mensagens de rastreamentoe, depois, escolhaAtivar.

   Você começará a receber mensagens de rastreamento em que a mensagem de rastreamento mais recente aparece primeiro no console.

   > **Note**
   >
   > Após o início da sessão de mensagens, receber mensagens de rastreamento pode incorrer em custos adicionais até que você desative a sessão ou saia da sessão de rastreamento. Para obter mais informações sobre a definição de preço, consulte Definição de preço do AWS IoT Core.

Exibir e monitorar mensagens de rastreamento

Depois de ativar as mensagens de rastreamento, o WebSocket a conexão é estabelecida e as mensagens de rastreamento começam a aparecer em tempo real, o mais novo primeiro. Você pode personalizar as preferências para especificar o número de mensagens de rastreamento a serem exibidas em cada página e exibir somente os campos relevantes para cada mensagem. Por exemplo, você pode personalizar o log de mensagens de rastreamento para mostrar somente logs para recursos de gateway sem fio que tenhamLog LevelDefinir como`ERROR`, para que você possa identificar e depurar rapidamente erros com seus gateways. As mensagens de rastreamento contêm as seguintes informações.

- Número da mensagem: Um número exclusivo que mostra a última mensagem recebida primeiro.
- Resource ID (ID do recurso): O gateway sem fio ou o ID do dispositivo sem fio do recurso.
- Time stamp: A hora em que a mensagem foi recebida.
- ID de mensagem: Um identificador queAWS IoT Corepara LorAwan atribui a cada mensagem recebida.
- FPort: A porta de frequência para se comunicar com o dispositivo usando o WebSocket Conexão.

- DevEui: O identificador exclusivo estendido (EUI) para o seu dispositivo sem fio.
- Recurso: Se o recurso monitorado é um dispositivo sem fio ou um gateway sem fio.
- Evento: O evento de uma mensagem de registro para um dispositivo sem fio, que pode serUnir,Junte-se novamente,Uplink_Data,Downlink_Data, ouRegistro.
- Log Level:Informações sobre`INFO`ou`ERROR`fluxos de registro para seu dispositivo.

Mensagem de log JSON do analisador de rede

Você também pode escolher uma mensagem de rastreamento de cada vez para exibir a carga JSON dessa mensagem. Dependendo da mensagem selecionada nos logs de mensagens de rastreamento, você verá informações na carga JSON que indica que contém 2 partes: Log do clienteeLoraFrame.

Log do cliente

OLog do clienteparte do JSON exibe o tipo e o identificador do recurso que recebeu a mensagem, o nível de log e o conteúdo da mensagem. O exemplo a seguir mostra umLog do clienteMensagem de log. Você pode usar `o``message`no JSON para obter mais informações sobre o erro e como ele pode ser resolvido.

LoraFrame

OLoraFrameparte do JSON tem umID de mensageme contém informações sobre a carga física para o dispositivo e os metadados sem fio.

A tabela a seguir mostra a estrutura da mensagem de rastreamento.

> **Note**
>
> Se seus dispositivos enviarem uma mensagem de uplink sem um valor para`Fport`,AWS IoT Corepara o analisador de rede LoRaWAN exibirá um`fPort`valor de 225 na mensagem de rastreamento recebida.

```
export type TraceMessage = {
  ResourceId: string;
  Timestamp: string;
  LoRaFrame:
  {
    MessageId: string;
    PhysicalPayload: any;
    WirelessMetadata:
    {
      fPort: number;
      dataRate: number;
      devEui: string;
      frequency: number,
      timestamp: string;
    },
  }
  CustomerLog:
  {
    resource: string;
    wirelessDeviceId: string;
    wirelessDeviceType: string;
    event: string;
    logLevel: string;
    messageId: string;
    message: string;
  },
};
```

Revisar e próximas etapas

Nesta seção, você visualizou mensagens de rastreamento e aprendeu como usar as informações para depurar erros. Depois de visualizar todas as mensagens, você pode:

- Desativar mensagens de rastreamento

  Para evitar incorrer em custos adicionais, você pode desativar a sessão de mensagens de rastreamento. Desativar a sessão desconecta o WebSocket conexão para que você não receba nenhuma mensagem de rastreamento adicional. Você ainda pode continuar exibindo as mensagens existentes no console.
- Editar informações de quadro para sua configuração

  Você pode editar a configuração do analisador de rede e escolher se deseja desativar as informações do quadro e escolher os níveis de log para suas mensagens. Antes de atualizar sua configuração, considere desativar a sessão de mensagens de rastreamento. Para fazer essas edições, abra oPágina de detalhes do Network Analyzer naAWS IoTconsolee escolhaEdite. Em seguida, você pode atualizar sua configuração com as novas configurações e ativar as mensagens de rastreamento para ver as mensagens atualizadas.
- Adicione recursos à configuração

  Você também pode adicionar mais recursos à configuração do analisador de rede e monitorá-los em tempo real. Você pode adicionar até um total combinado de 250 recursos de dispositivos sem fio e gateway sem fio. Para adicionar recursos, noPágina de detalhes do Network Analyzer doAWS IoTconsole, escolha oRecursosguia e escolhaAdicione recursos. Em seguida, você pode atualizar sua configuração com os novos recursos e ativar as mensagens de rastreamento para ver as mensagens atualizadas para os recursos adicionais.

Para obter mais informações sobre como atualizar a configuração do analisador de rede editando as configurações e adicionando recursos, consulteAdicione recursos e atualize a configuração do analisador de rede (p. 1207).

Segurança de dados com oAWS IoT Corepara LoRaWAN

Dois métodos protegem os dados do seuAWS IoT Corepara dispositivos LoRaWAN:

- A segurança que os dispositivos sem fio usam para se comunicar com os gateways.

  Os dispositivos LoRaWAN seguem as práticas de segurança descritas emSEGURANÇA LOraWAN™: Um white paper preparado para o LoRa Alliance™ de Gemalto, Actility e Semtechpara se comunicar com os gateways.
- A segurança doAWS IoT Coreusa para conectar gateways aAWS IoT Corepara LoraWAN e envie os dados para outrosAWSServiços da .

  AWS IoT Coresegurança é descrita emProteção de dados no AWS IoT Core (p. 380).

Como os dados são protegidos em todo o sistema

Este diagrama identifica os elementos-chave em um sistema LoRawan conectado aAWS IoT Corepara LoraWAN identificar como os dados são protegidos por toda parte.

1. O dispositivo sem fio LoRaWAN criptografa suas mensagens binárias usando o modo AES128 CTR antes de transmiti-las.
2. Conexões de gateway para oAWS IoT Corepara LoRaWAN são protegidos pelo TLS conforme descrito emSegurança de transporte no AWS IoT (p. 381).AWS IoT Corefor LoRaWAN descriptografa a mensagem binária e codifica a carga útil da mensagem binária descriptografada como uma string base64.
3. A mensagem codificada em base64 resultante é enviada como a carga útil da mensagem para oAWS IoTregra descrita no destino atribuído ao dispositivo. Dados dentroAWSé criptografado usandoAWS-chaves de propriedade.
4. OAWS IoTregra direciona os dados da mensagem para os serviços descritos na configuração da regra. Dados dentroAWSé criptografado usandoAWS-chaves de propriedade.

Segurança de transporte de dispositivos e gateway LoRaWAN

Dispositivos LoRaWAN eAWS IoT Corepara armazenar chaves raiz pré-compartilhadas da LoRaWAN. As chaves de sessão são derivadas por dispositivos LoraWAN eAWS IoT Corepara LorAwan seguindo os protocolos. As chaves de sessão simétricas são usadas para criptografia e descriptografia em um modo AES-128 CTR padrão. Um código de integridade de mensagem (MIC) de 4 bytes também é usado para verificar a integridade dos dados seguindo um algoritmo padrão AES-128 CMAC. As chaves de sessão podem ser atualizadas usando o processo Entrar/Reingressar.

A prática de segurança para LoRa gateways são descritos nas especificações da LoRaWAN. LoRa Gateways se conectam aAWS IoT Corepara LoRaWAN através de um soquete da web usando um`Basics Station`.AWS IoT Corepara LoRaWAN apenas para o LoRaWAN`Basics Station`versão 2.0.4 e posterior.

Antes que a conexão do soquete da web seja estabelecida,AWS IoT Corepara LoRaWAN usa oModo de Autenticação de Cliente e Servidor TLS (p. 381)para autenticar o gateway.AWS IoT Corepara LoRaWAN também mantém um Servidor de Configuração e Atualização (CUPS) que configura e atualiza os certificados e chaves usados para autenticação TLS.

Integração com o Amazon Sidewalk paraAWS IoT Core

Amazon Sidewalké uma rede compartilhada que ajuda dispositivos como Amazon Echo, câmeras de segurança Ring, luzes externas, sensores de movimento e rastreadores de azulejos a trabalhar melhor em casa e além da porta da frente. Quando ativada, essa rede pode oferecer suporte a outros dispositivos Sidewalk em sua comunidade e abrir as portas para inovações, como localizar itens conectados ao Amazon Sidewalk. O Amazon Sidewalk ajuda seus dispositivos a se conectarem e permanecerem conectados. Por exemplo, se o dispositivo perder a conexão Wi-Fi, o Sidewalk pode simplificar a reconexão ao roteador. Para obter mais informações, consulteGuia de início rápido do Amazon Sidewalk.

As seções a seguir mostram como integrar seus dispositivos Sidewalk comAWS IoTe use notificações de eventos para notificá-lo de eventos como quando seu dispositivo Sidewalk está registrado. Para obter informações sobre como usar a Amazon CloudWatch para monitorar seus dispositivos Sidewalk, consulteMonitorar e registrar em log paraAWS IoT WirelessUsar o Amazon CloudWatch (p. 1226).

Como integrar seu dispositivo Sidewalk

Você pode integrar seus dispositivos Sidewalk paraAWS IoTUsando o console ou oAWS IoTAPI Wireless. Depois que seus dispositivos Amazon Sidewalk forem autenticados, suas mensagens são enviadas paraAWS IoT Core. Em seguida, você pode começar a desenvolver seus aplicativos de negócios noAWSCloud, que usa os dados de seus dispositivos Amazon Sidewalk.

Usar o console

Para integrar seus dispositivos Sidewalk usando oAWS Management ConsolePrimeiro registre seu dispositivo noConsole Sidewalk Developer Service (SDS)conta e, em seguida, associe sua Amazon ID à suaConta da AWS. Para ver os dispositivos Sidewalk que você adicionou e gerenciá-los, faça login noAWS Management Consolee navegue até oDispositivosPágina noAWS IoTconsole do .

Usando a API ou a CLI

Você pode integrar os dispositivos Sidewalk e LoRawan usando oAWS IoTSem fioAPI. OAWS IoTAPI WirelessAWS IoT Coreé construído em é suportado peloAWSSDK. Para obter mais informações, consulteAWSSDKs e Toolkits.

Você pode usar oAWS CLIpara executar comandos para integração e gerenciamento de seus dispositivos Sidewalk. Para obter mais informações, consulteAWS IoTReferência da CLI do.

Dispositivos Sidewalk integrados com Amazon Sidewalk Integration paraAWS IoT Core

Com a integração do Amazon Sidewalk paraAWS IoT Core, você pode adicionar sua frota de dispositivos Sidewalk aoAWSNuvem. Use as seguintes tarefas para começar a usar o.

1. Revise o SDK do Sidewalk e a documentação

   Saiba mais sobre o Amazon Sidewalk e como seus dispositivos podem usá-lo.

   a. Consulte o Guia de início rápido do Amazon Sidewalk.

   b. Faça o download de um SDK para Amazon Sidewalk.
   c. Abrir oConsole Sidewalk Developer Service (SDS).
2. Registre seu protótipo de dispositivo

   No console do SDS, registre seu protótipo de dispositivo com o Amazon Sidewalk.
3. Associe seu ID da Amazon Sidewalk à suaConta da AWS

   NoAWS IoTconsole, associe seu ID da Amazon Sidewalk à suaConta da AWS.

   Seus dispositivos Amazon Sidewalk aparecem naCalçadaGuia doDispositivosHub doAWS IoTconsole.
4. Conclua a configuração do dispositivo Amazon Sidewalk naAWS IoTconsole

   Crie os destinos e as regras que seu dispositivo Sidewalk precisa para rotear e formatar os dados paraAWSServiços da .

Os tópicos a seguir mostram como você pode adicionar dispositivos Sidewalk e conectá-los aAWS IoT. Antes de adicionar seus dispositivos, verifique se oConta da AWSTem as permissões necessárias do IAM para executar o seguinteprocedimentos.

Tópicos

- Adicione as credenciais da sua conta do Sidewalk (p. 1220)
- Adicionar um destino para o seu dispositivo Sidewalk (p. 1221)
- Criar regras para processar mensagens do dispositivo Sidewalk (p. 1223)

Adicione as credenciais da sua conta do Sidewalk

Você pode conectar dispositivos Sidewalk aoAWS IoTusando oAWS Management Consoleou oAWS IoTAPI Wireless. Para integrar seu dispositivo, criaremos um perfil de conectividade sem fio para o seu dispositivo Sidewalk e, em seguida, adicionaremos um destino eAWS IoTregra para os pontos de extremidade do perfil e da calçada.

Antes de adicionar o dispositivo, você deve adicionar as credenciais da conta do Sidewalk. Você pode adicionar suas credenciais usando oAWS Management Consoleou oAWS IoTAPI Wireless.

Adicione suas credenciais de conta do Sidewalk usando o console

Para adicionar as credenciais da conta do Sidewalk a partir do console:

1. Navegue até o .PerfisPágina doAWS IoTConsole do e escolha oCalçadaGuia.

   Note

   Certifique-se de que você esteja usando `ous-east-1`Região : Essa guia não será exibida no console se você estiver usando uma região diferente.
2. Insira o ID da Amazon Sidewalk. Você obtém esse ID doConsole Sidewalk Developer Service (SDS)ao projetar seu produto Sidewalk. Para obter mais informações, consulteElabore o seu produto Sidewalk.
3. Carregar oAppServerPrivateKey, que é a chave do servidor que seu fornecedor forneceu. OAppServerPrivateKeyé a chave privada ED25519 (`oapp-server-ed25519-private.txt`file), que é um valor hexadecimal de 64 dígitos. Você gerou essa chave usando a ferramenta de geração de certificados Sidewalk quando projetou seu produto Sidewalk. Para obter mais informações, consulteElabore o seu produto Sidewalk.
4. Para adicionar suas credenciais do Sidewalk, escolhaAdicionar credencial.

Adicione suas credenciais de conta do Sidewalk usando a API

Você pode usar oAWS IoTAPI sem fio para adicionar as credenciais da conta do Sidewalk. A lista a seguir descreve as ações da API.

AWS IoTAções da API sem fio para a conta Sidewalk

- AssociateAWSACCountWithPartnerAccount
- DisasociateAWSACCOUNTFromPartnerAccount
- GetPartnerAccount
- ListPartnerAccounts
- updatePartnerAccount

Para obter a lista completa de ações e tipos de dados disponíveis para criar e gerenciarAWS IoT CorePara obter recursos LoRaWAN, consulte oAWS IoTReferência da API do.

Como usar aAWS CLIPara adicionar uma conta

Você pode usar oAWS CLIpara associar uma conta do Sidewalk à suaConta da AWSusando oassociate-aws-account-with-partner-account, conforme ilustrado pelo exemplo a seguir.

> **Note**
>
> Também é possível executar esse procedimento com a API usando os métodos na API da AWS que correspondam aos comandos da CLI mostrados aqui.

```
aws iotwireless associate-aws-account-with-partner-account \
    --sidewalk
 AmazonId="12345678901234",AppServerPrivateKey="a123b45c6d78e9f012a34cd5e6a7890b12c3d45e6f78a1b234c56d7
```

Próximas etapas

Agora que você adicionou as credenciais e configurou um perfil de conectividade sem fio, você pode adicionar um destino ao seu dispositivo Sidewalk. Você verá as credenciais que você adicionou naPerfisPágina doAWS IoTConsole do, noCalçadaGuia. Você definirá um nome de função e um nome de regra para o destino que pode rotear mensagens enviadas de seus dispositivos. Para obter mais informações, consulte Adicionar um destino para o seu dispositivo Sidewalk (p. 1221).

Adicionar um destino para o seu dispositivo Sidewalk

Antes que você possa adicionar umAWS IoT Corepara o destino LoRaWAN e crie uma regra para rotear as mensagens enviadas do seu dispositivo Sidewalk, você deve criar um perfil de conectividade sem fio. Para criar o perfil, primeiro registre seu dispositivo Sidewalk e, em seguida, adicione as credenciais ao seuConta da AWS. Para obter mais informações, consulte Adicione as credenciais da sua conta do Sidewalk (p. 1220).

Criar um destino de calçada é semelhante à forma como você cria um destino para seus dispositivos LoRaWAN. A seguir, mostra como você pode criar um destino usando oAWS Management Consoleou a API.

Em Adicionar um destino usando o console

Você pode adicionar seu destino Sidewalk a partir doDestinosPágina doAWS IoTconsole do .

Especifique os seguintes campos ao criar umAWS IoT Corepara o destino LoRaWAN e, em seguida, escolhaAdicionar destino.

- Detalhes do destino

  Insira umNome do destinoe uma descrição opcional para seu destino. Para oNome do destino, insira**`SidewalkDestination`**. Opcionalmente, insira uma descrição, como**`This is a destination for Sidewalk devices`**.
- Nome da regra

  OAWS IoTregra configurada para processar os dados do dispositivo. Seu destino precisa de uma regra para processar as mensagens recebidas. Digite um nome de regra (digamos**`SidewalkRule`**) e, depois, escolhaCopiarpara copiar o nome da regra que você inserirá ao criar oAWS IoTregra. Você também pode escolherCriar regrapara criar a regra agora ou navegar até oRegrasHub doAWS IoTconsole e crie uma regra com o nome que você copiou.

  Para obter mais informações sobreAWS IoTregras para destinos, consulteCriar regras para processar mensagens do dispositivo Sidewalk.
- Nome da função

  A função do IAM que dá permissão aos dados do dispositivo para acessar a regra nomeada emRule name (Nome da regra). Para criar a função do IAM, siga as etapas descritas emCriar uma função do IAM para seus destinos (p. 1145). Ao criar a função:
  - para oSelecione o tipo de entidade confiável, escolhaAWS service (Serviço da AWS)e, depois, escolhaIoTcomo o serviço.
  - Digite**`SidewalkRole`**para aNome da função.
  - Use o mesmo documento de política conforme descrito emCriar uma função do IAM para seus destinos (p. 1145).

  Para obter mais informações sobre funções do IAM, consulteUso de funções do IAM.

Adicione um destino usando a API

As listas a seguir descrevem as ações da API que executam as tarefas associadas à adição, atualização ou exclusão de um destino.

AWS IoTAções de API sem fio para perfis de serviço

- CriadoEstinação
- GetDestination
- ListDestinations
- UpdateDestination
- DeleteDestination

Para obter a lista completa de ações e tipos de dados disponíveis para criar e gerenciarAWS IoT CorePara obter recursos LoRaWAN, consulte oAWS IoTReferência da API do.

Como usar aAWS CLIPara adicionar um destino

Você pode usar oAWS CLIpara adicionar um destino usando ocriar-destinocomando. O exemplo a seguir cria um destino.

```
aws iotwireless create-destination \
    --name SidewalkDestination \
    --expression-type RuleName \
    --expression SidewalkRule \
    --role-arn arn:aws:iam::123456789012:role/SidewalkRole
```

A execução desse comando cria um destino com o nome de destino, o nome da regra e o nome da função especificados. Para obter informações sobre nomes de regras e funções para destinos, consulteCriar regras para processar mensagens do dispositivo Sidewalk.

Para obter informações sobre as CLIs que podem ser usadas, consulteAWS CLIreferência.

Próximas etapas

Agora que você adicionou o destino, você pode criar a regra de destino para o seu dispositivo Sidewalk que roteará mensagens para outros serviços. Para obter mais informações, consulte Criar regras para processar mensagens do dispositivo Sidewalk (p. 1223).

Criar regras para processar mensagens do dispositivo Sidewalk

AWS IoTas regras podem receber as mensagens dos dispositivos Sidewalk e encaminhá-las para outros serviços.AWS IoT Corepara destinos LoRaWAN (p. 1144)associe um dispositivo Sidewalk à regra que processa os dados da mensagem do dispositivo para enviar a outros serviços.

Você pode usar uma regra existente para seu destino. Nesta seção, criaremos a regra,`SidewalkRule`, que você especificou ao criar o destino Sidewalk, conforme descrito emAdicionar um destino para o seu dispositivo Sidewalk (p. 1221). Ao criar a regra, criaremos umAWS LambdaAção para publicar novamente a mensagem em umAWS IoTtópico.

Criar uma regra de destino na calçada

Navegue até o .RegrasHub doAWS IoTConsole e execute as seguintes etapas.

1. SelecioneCriar uma regraPara criar uma nova regra para o destino.
2. Insira o nome**`SidewalkRule`**para aName (Nome)e especifique um opcionalDescriçãopara a regra (por exemplo,**`Sidewalk rule for lambda action to republish a topic`**).
3. Alterar a instrução de consulta padrão para**`SELECT *`**Para que todas as ações associadas à regra sejam executadas. Mantenha a versão SQL para`2016-03-23`.
4. Em Set one or more actions (Definir uma ou mais ações), selecione Add action (Adicionar ação).
5. Para a ação da regra, selecioneEnvie uma mensagem para uma função do LambdaE, depois, escolhaConfigure action.
6. Você pode escolher uma função do Lambda existente ou criar uma nova. Neste exemplo, criaremos uma função do Lambda. SelecioneCriar uma nova função Lambda.

Crie sua função usandoAWS Lambda

EscolhendoCriar uma nova função LambdaAbre aFunçõesPágina do console do Lambda. Siga estas etapas.

1. Para criar sua função, selecioneAuthor from scratch.
2. para oNome da função, digite um nome (por exemplo,**`Sidewalk_Handler`**), escolha`Python 3.8`comoTempo de execuçãoe, depois, escolhaCriar função.
3. Selecione oArquivo lambda.pyfunção noFonte do códigoSeção do console do.
4. No corpo da função, exclua qualquer código dentro do corpo da função e adicione uma instrução de impressão para sua função do Lambda. Você também pode usar base64 para decodificar o`PayloadData`para receber os dados do aplicativo para os quais o dispositivo enviaAWS IoT. A seguir, mostra um exemplo de função do Lambda.

```
import json
```

```
import base64

def lambda_handler(event, context):

    message = json.dumps(event)
    print (message)

    payload_data = base64.b64decode(event["PayloadData"])
    print(payload_data)
    print(int(payload_data,16))
```

5. Para implantar o código da função, escolhadisponibilizar.
6. Retorno para oRegrasConclua o console e atualize a página. Escolha a função do Lambda que você criou e escolhaAdicionar ação.

Republique uma mensagem em umAWS IoTtópico

Você pode adicionar uma segunda ação para republicar uma mensagem em umAWS IoTtópico doRegrasHub do console.

1. Selecione Add action.
2. SelecioneRepublique uma mensagem em umAWS IoTtópicoe escolhaConfigure action.
3. Digite**`project/sensor/observed`**para aTópicoE certifique-se de que oQualidade do serviçoé definido como0 - A mensagem é entregue zero ou mais vezes.
4. Selecione Create Role. DigiteSidewalkRepublishRolePara o nome da função e escolhaCriar função.
5. Selecione Add action.

   Ambas as ações aparecem noRegrasHub doAWS IoTconsole do .
6. Selecione Create rule (Criar regra).

   A regra aparece noRegrasPágina que mostra a lista de regras.

Próximas etapas

Agora que você criou a regra de destino para o seu dispositivo Sidewalk, você pode conectar seu dispositivo e observar mensagens sobre o tópico que você se inscreveu. Para obter mais informações, consulte Connect seu dispositivo Sidewalk e visualize o formato de metadados de uplink (p. 1224).

Connect seu dispositivo Sidewalk e visualize o formato de metadados de uplink

Depois de adicionar suas credenciais do Sidewalk e adicionar o destino, você pode provisionar seus endpoints do Sidewalk e conectar seu dispositivo.

Connect o dispositivo Sidewalk

Você pode provisionar seu dispositivo como um endpoint do Sidewalk gerando os certificados de dispositivo e os certificados do servidor de aplicativos a partir doConsole Sidewalk Developer Service (SDS). Para obter mais informações, consulteProvisione e configure seus endpoints de calçada.

Depois de conectar o dispositivo, você verá o dispositivo Sidewalk noDispositivosPágina doAWS IoTConsole do, noCalçadaGuia. Quando o dispositivo estiver conectado e começar a enviar dados, você verá a data e a hora doÚltimo uplink recebido emcampo.

Exibir formato das mensagens de uplink

Depois de conectar o dispositivo, você pode se inscrever no tópico (por exemplo,`project/sensor/observed`) que você especificou ao criar a regra de destino Sidewalk e observar mensagens de uplink do dispositivo. Para se inscrever no tópico, vá para oCliente de teste MQTTnoTestePágina doAWS IoTConsole, digite o nome do tópico (por exemplo,`project/sensor/observed`) e, depois, escolhaSubscribe.

O exemplo a seguir mostra o formato das mensagens de uplink que são enviadas de dispositivos Sidewalk paraAWS IoT. O`WirelessMetadata`Contém metadados sobre a solicitação de mensagem.

```
{
   "PayloadData":"ZjRlNjY1ZWNlNw==",
   "WirelessDeviceId":"6dc562bf-31c5-37e0-b230-716ea8148c3d",
   "TransmitMode": "0",
   "WirelessMetadata":{
      "Sidewalk":{
         "CmdExStatus":"Cmd",
         "SidewalkId":"device-id",
         "Seq":0,
         "MessageType":"messageType"
      }
```

A tabela a seguir mostra uma definição dos diferentes parâmetros nos metadados de uplink. O*`device-id`*É o ID do dispositivo sem fio, como*`ABCDEF1234`*O e a*`messageType`*é o tipo de mensagem de upllink recebida do dispositivo.

Parâmetros de metadados de uplink de calçada

Parâmetro	Descrição	Type	Obrigatório
`PayloadData`	A carga útil da mensagem enviada do dispositivo sem fio.	String	Sim
`WirelessDeviceID`	O identificador do dispositivo sem fio que está enviando os dados	String	Sim
`TransmitMode`	O modo de transmissão para dados enviados do dispositivo sem fio. Pode ser`0`pelo`unconfirmed`Modo do,`1`pelo`confirmed`, e`2`pelo`unused`.	Inteiro	Sim
`Sidewalk.CmdExStatus`	Status do tempo de execução do comando. As mensagens do tipo de resposta devem incluir o código de status,`COMMAND_EXEC_STATUS_SUCCESS`. No entanto, as notificações podem não incluir o código de status.	Enumeração	Não
`Sidewalk.NackExStatus`	Status de nack de resposta, que pode ser`RADIO_TX_ERROR`ou`MEMORY_ERROR`.	Matriz de strings	Não

Monitorar e registrar em log paraAWS IoT WirelessUsar o Amazon CloudWatch

Você pode monitorar seuAWS IoT Wirelessrecursos e aplicativos que são executados em tempo real usando o Amazon CloudWatch. Você pode monitorar o status de seus dispositivos LoraWAN e Sidewalk que você integrou usando o Amazon CloudWatch.

- Para obter informações sobre como integrar-se dispositivos LoRawan, consulteConectando gateways e dispositivos aAWS IoT Core for LoRaWAN (p. 1130).
- Para obter informações sobre como integrar-se dispositivos do Amazon Sidewalk paraAWS IoT Core, consulteDispositivos Sidewalk integrados com Amazon Sidewalk Integration paraAWS IoT Core (p. 1219).

Usar o CloudWatch Para coletar e monitorar métricas, que são as variáveis que é possível medir para avaliar seus recursos e aplicativos. Para obter mais informações sobre os benefícios de usar o monitoramento, consulteMonitorar o AWS IoT (p. 432).

Se você quiser obter mais informações de log em tempo real dos seus dispositivos LoRaWAN, use o analisador de rede. Para obter mais informações, consulteMonitorando sua frota de recursos sem fio em tempo real usando o analisador de rede (p. 1201)

Como monitorar seus recursos sem fio

Para registrar e monitorar seus recursos sem fio, siga as etapas a seguir.

1. Criar uma função de registro em log para registrar oAWS IoT Wirelessrecursos, conforme descrito emCriar uma política e uma função de registro em log paraAWS IoT Wireless (p. 1227).
2. Mensagens de log no CloudWatch O console de registros tem um nível de log padrão de`ERROR`, que é menos detalhado e contém somente informações de erros. Se você quiser visualizar mensagens mais detalhadas, recomendamos que você use a CLI para configurar o registro em log primeiro, como descrito emConfigurar o registro em log para oAWS IoT Wirelessrecursos (p. 1229).
3. Em seguida, você pode monitorar seus recursos visualizando as entradas de registro na CloudWatch Console de logs. Para obter mais informações, consulteExibir CloudWatch AWS IoT WirelessEntradas de log do do (p. 1238)
4. Você pode criar expressões de filtro usandoGrupos de logsmas recomendamos que você primeiro crie filtros simples e visualize entradas de registro nos grupos de log e, em seguida, vá para CloudWatch Insights para criar consultas para filtrar as entradas de registro, dependendo do recurso ou evento que você está monitorando. Para obter mais informações, consulteUsar o CloudWatch Insights para filtrar logs paraAWS IoT Wireless (p. 1243)

Os tópicos a seguir mostram como configurar o registro em log paraAWS IoT Wirelesse para coletar métricas do CloudWatch. Além dos dispositivos LoRaWAN, você pode usar esses tópicos para configurar o registro em log para qualquer dispositivo Sidewalk que você adicionou à sua conta e monitorá-los. Para obter informações sobre como adicionar esses dispositivos, consulteIntegração com o Amazon Sidewalk paraAWS IoT Core (p. 1219).

Tópicos

- Configurar o registro em log para o AWS IoT Wireless (p. 1227)

- MonitorAWS IoT Wirelessusando CloudWatch Logs (p. 1237)

Configurar o registro em log para o AWS IoT Wireless

Antes que você possa monitorar e registrarAWS IoTatividade, primeiro habilite o registro em log paraAWS IoT Wirelessrecursos usando a CLI ou a API.

Ao considerar como configurar oAWS IoT Wirelesslog, a configuração de log padrão determina comoAWS IoTA atividade será registrada a menos que você especifique o contrário. A partir do, talvez você queira obter logs detalhados com um nível de log padrão de`INFO`.

Depois de revisar os logs iniciais, você pode alterar o nível de log padrão para`ERROR`, que é menos detalhado, e define um nível de log mais detalhado e específico de recursos em recursos que possam precisar de mais atenção. É possível alterar os níveis de log sempre que quiser.

Os tópicos a seguir mostram como configurar o registro em log paraAWS IoT Wirelessrecursos da AWS.

Tópicos

- Criar uma política e uma função de registro em log paraAWS IoT Wireless (p. 1227)
- Configurar o registro em log para oAWS IoT Wirelessrecursos (p. 1229)

Criar uma política e uma função de registro em log paraAWS IoT Wireless

A seguir mostra como criar uma função de registro em log para somenteAWS IoT Wirelessrecursos da AWS. Se você também quiser criar uma função de registro paraAWS IoT Core, consulteCriar uma função de registro em log (p. 433).

Criar uma função de registro em log paraAWS IoT Wireless

Antes que possa habilitar o registro em log, você deve criar uma função do IAM e uma política que forneçaAWSpermissão para monitorarAWS IoT WirelessAtividade em seu nome.

Criar função do IAM para o registro em log

Para criar uma função de registro em log paraAWS IoT Wireless, abra aHub de funções do console do IAMe escolhaCriar função.

1. UnderSelecione o tipo de entidade confiável, escolhaOutroAWSconta.
2. DentroID da conta, insira seuAWSID da conta e, em seguida, escolhaPróximo: Permissions
3. Na caixa de pesquisa, insira **`AWSIoTWirelessLogging`**.
4. Selecione a caixa ao lado da política nomeadaAWSIoTWirelessLogginge, depois, escolhaPróximo: Tags.
5. Selecione Next (Próximo): Análise.
6. DentroNome da função, insira**`IoTWirelessLogsRole`**e, depois, escolhaCriar função.

Editar relação de confiança da função do IAM

Na mensagem de confirmação exibida após a execução da etapa anterior, escolha o nome da função que você criou,IOTWirelessLogsRole. Em seguida, você editará a função para adicionar o seguinte relacionamento de confiança.

1. NoSummary (Resumo)seção da funçãoIOTWirelessLogsRole, escolha oRelações de confiançae, em seguida, escolhaEditar relação de confiança.
2. DentroDocumento da política, altere o`Principal`Propriedade para ser semelhante a este exemplo.

```
"Principal": {
    "Service": "iotwireless.amazonaws.com"
},
```

   Depois de alterar o`Principal`A propriedade, o documento de política completo deve ser semelhante a este exemplo.

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

3. Para salvar as alterações e sair do, escolhaAtualizar política de confiança.

Política de registro em log paraAWS IoT Wireless

O documento de políticas a seguir fornece a política de função e de confiança que permiteAWS IoT WirelessPara enviar entradas de log para CloudWatch Em seu nome.

> **Note**
>
> EsseAWSO documento de política gerenciado foi criado automaticamente para você quando você criou a função de log,IOTWirelessLogsRole.

Política de função

A seguir, mostra o documento de política de função.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogGroup",
                "logs:CreateLogStream",
                "logs:DescribeLogGroups",
                "logs:DescribeLogStreams",
                "logs:PutLogEvents"
            ],
            "Resource": "arn:aws:logs:*:*:log-group:/aws/iotwireless*"
        }
    ]
}
```

Política de confiança para registrar somenteAWS IoT Wirelessatividade

A seguir, mostra a política de confiança somente para registro em logAWS IoT Wirelessatividade.

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

Se você criou a função do IAM para registrar tambémAWS IoT Coreatividade e, em seguida, os documentos de política permitem que você registre ambas as atividades. Para obter informações sobre como criar uma função de registro em log paraAWS IoT Core, consulteCriar uma função de registro em log (p. 433).

Next steps (Próximas etapas)

Você aprendeu a criar uma função de registro em log para registrar seuAWS IoT Wirelessrecursos da AWS. Por padrão, os logs têm um nível de log de`ERROR`, então, se você quiser ver apenas informações de erro, vá paraExibir CloudWatch AWS IoT WirelessEntradas de log do do (p. 1238)para monitorar seus recursos sem fio exibindo as entradas de registro.

Se você quiser mais informações nas entradas de registro, você pode configurar o nível de log padrão para seus recursos ou para diferentes tipos de evento, como definir o nível de log como`INFO`. Para obter informações sobre como configurar o registro em log para seus recursos, consulteConfigurar o registro em log para oAWS IoT Wirelessrecursos (p. 1229).

Configurar o registro em log para oAWS IoT Wirelessrecursos

Para configurar o registro em log paraAWS IoT WirelessOs recursos do, você pode usar a API ou a CLI. Ao começar a monitorarAWS IoT WirelessRecursos, você pode usar a configuração padrão. Para fazer isso, você pode pular este tópico e prosseguir paraMonitorAWS IoT Wirelessusando CloudWatch Logs (p. 1237)para monitorar seus registros.

Depois de começar a monitorar os logs, você pode usar a CLI para alterar os níveis de log para uma opção mais detalhada, como fornecer`INFO`e`ERROR`informações e habilitar o registro em log para mais recursos.

AWS IoT WirelessRecursos e níveis de log

Antes de usar a API ou a CLI, use a tabela a seguir para saber mais sobre os diferentes níveis de log e os recursos para os quais você pode configurar o log. A tabela mostra os parâmetros que você vê na CloudWatch registra quando você monitora os recursos. A forma como você configura o registro em log para seus recursos determinará os logs que você vê no console.

Para obter informações sobre o que uma amostra CloudWatch logs se parece e como você pode usar esses parâmetros para registrar informações úteis sobre oAWS IoT WirelessRecursos, consulteExibir CloudWatch AWS IoT WirelessEntradas de log do do (p. 1238).

Níveis e recursos de log

Name (Nome)	Possíveis valores	Descrição
`logLevel`	`INFO`, `ERROR` ou `DISABLED`	• `ERROR`: Exibe qualquer erro que cause a falha de uma operação. Os registros incluem apenas`ERROR`Informações. • `INFO`: Fornece informações de alto nível sobre o fluxo de coisas. Os registros incluem`INFO`e`ERROR`Informações. • `DISABLED`: Desabilita todo o registro em log.
`resource`	`WirelessGateway` ou `WirelessDevice`	O tipo de recurso, que pode ser`WirelessGateway`ou`WirelessDevice`.
`wirelessGatewayType`	`LoRaWAN`	O tipo de gateway sem fio, quando`resource`é`WirelessGateway`, que é sempre o LorAwan.
`wirelessDeviceType`	`LoRaWAN` ou `Sidewalk`	O tipo de dispositivo sem fio, quando`resource`é`WirelessDevice`, que pode ser`LoRaWAN`ou`Sidewalk`.
`wirelessGatewayId`	-	O identificador do gateway sem fio, quando`resource`é`WirelessGateway`.
`wirelessDeviceId`	-	O identificador do dispositivo sem fio, quando`resource`é`WirelessDevice`.
`event`	`Join`,`Rejoin`,`Registration`,`Update`,`Data`,`Command`,`Certificate`	O tipo de evento que está sendo registrado, que depende se o recurso que você está registrando é um dispositivo sem fio ou um gateway sem fio. Para obter mais informações, consulteExibir CloudWatch AWS IoT WirelessEntradas de log do do (p. 1238)

AWS IoT WirelessAPI de registro

Você pode usar as ações da API do a seguir para configurar o registro em log de recursos. A tabela também mostra um exemplo de política do IAM que você deve criar para usar as ações da API. A seção a seguir descreve como usar as APIs para configurar níveis de log de seus recursos.

Ações de API de log

Nome da API	Descrição	Exemplo de política do IAM
GetLogLevelsByResourceTypes	Retorna os níveis de log padrão atuais ou níveis de log por tipos de recursos, que podem incluir opções de log para dispositivos sem fio ou gateways sem fio.	`{ "Version": "2012-10-17", "Statement": [ { "Effect": "Allow", "Action": [ "iotwireless:GetLogLevelsByResourceTypes ],`

Nome da API	Descrição	Exemplo de política do IAM
		`"Resource": [ "*" ] } ] }`
GetResourceLogLevel	Retorna a substituição no nível de log para um determinado identificador de recurso e tipo de recurso. O recurso pode ser um dispositivo sem fio ou um gateway sem fio.	`{ "Version": "2012-10-17", "Statement": [ { "Effect": "Allow", "Action": [ "iotwireless:GetResourceLogLevel" ], "Resource": [ "arn:aws:iotwireless:us-east-1:123456789012:WirelessDevice/012bc5ab12-cd3a-d00e-1f0e20c1204a", ] } ] }`
PutResourceLogLevel	Define a substituição no nível de log para um determinado identificador de recurso e tipo de recurso. O recurso pode ser um gateway sem fio ou um dispositivo sem fio. Note Essa API tem um limite de 200 substituições no nível de log por conta.	`{ "Version": "2012-10-17", "Statement": [ { "Effect": "Allow", "Action": [ "iotwireless:PutResourceLogLevel" ], "Resource": [ "arn:aws:iotwireless:us-east-1:123456789012:WirelessDevice/012bc5ab12-cd3a-d00e-1f0e20c1204a", ] } ] }`

Nome da API	Descrição	Exemplo de política do IAM
ResetallResourceLogLevels	Remove as substituições no nível de log para todos os recursos, o que inclui gateways sem fio e dispositivos sem fio. Note Essa API não afeta os níveis de log definidos usando `oUpdateLogLevelsByResourceTypes`API.	`{ "Version": "2012-10-17", "Statement": [ { "Effect": "Allow", "Action": [ "iotwireless:ResetAllResourceLogLevels" ], "Resource": [ "arn:aws:iotwireless:us-east-1:123456789012:WirelessDevice/*", "arn:aws:iotwireless:us-east-1:123456789012:WirelessGateway/* ] } ] }`
ResetResourceLogLevel	Remove a substituição no nível de log para um determinado identificador de recurso e tipo de recurso. O recurso pode ser um gateway sem fio ou um dispositivo sem fio.	`{ "Version": "2012-10-17", "Statement": [ { "Effect": "Allow", "Action": [ "iotwireless:ResetResourceLogLevel" ], "Resource": [ "arn:aws:iotwireless:us-east-1:123456789012:WirelessDevice/012bc5ab12-cd3a-d00e-1f0e20c1204a", ] } ] }`

Nome da API	Descrição	Exemplo de política do IAM
UpdateLogLevelsByResourceTypes	Defina níveis de log padrão ou log por tipos de recurso. Você pode usar essa API para opções de log para dispositivos sem fio ou gateways sem fio e controlar as mensagens de registro que serão exibidas no CloudWatch. Note Os eventos são opcionais e o tipo de evento está vinculado ao tipo de recurso. Para obter mais informações, consulteEventos e tipos de recursos (p. 1238)	`{ "Version": "2012-10-17", "Statement": [ { "Effect": "Allow", "Action": [ "iotwireless:UpdateLogLevelsByResourceTy ], "Resource": [ "*" ] } ] }`

Configurar níveis de log de recursos usando a CLI

Esta seção descreve como configurar níveis de log paraAWS IoT Wirelessrecursos usando a API ouAWS CLI.

Antes de usar a CLI:

- Certifique-se de ter criado a política do IAM para a API para a qual deseja executar o comando da CLI, conforme descrito anteriormente.
- É necessário o nome de recurso da Amazon (ARN) da função que você deseja usar. Se você precisar criar uma função a ser usada para o registro em log, consulteCriar uma política e uma função de registro em log paraAWS IoT Wireless (p. 1227).

Por que usar oAWS CLI

Por padrão, se você criar a função do IAM,`IoTWirelessLogsRole`, conforme descrito emCriar uma política e uma função de registro em log paraAWS IoT Wireless (p. 1227), você verá CloudWatch logs noAWS Management Consoleque têm um nível de log padrão de`ERROR`. Para alterar o nível de log padrão para todos os seus recursos ou para recursos específicos, use oAWS IoT WirelessAPI de log ou CLI.

Como usar oAWS CLI

As ações da API podem ser categorizadas nos seguintes tipos, dependendo se você deseja configurar os níveis de log para todos os recursos ou para recursos específicos:

- Ações da API do`GetLogLevelsByResourceTypes`e`UpdateLogLevelsByResourceTypes`pode recuperar e atualizar os níveis de registro para todos os recursos em sua conta que são de um tipo específico, como um gateway sem fio ou um dispositivo LoRaWAN ou Sidewalk.
- Ações da API do`GetResourceLogLevel`,`PutResourceLogLevel`, e`ResetResourceLogLevel`pode recuperar, atualizar e redefinir os níveis de log de recursos individuais especificados por você usando um identificador de recurso.

- Ação de API`ResetAllResourceLogLevels`redefine a substituição no nível de log para`null`para todos os recursos para os quais você especificou uma substituição no nível de log usando `oPutResourceLogLevel`API.

Como usar a CLI para configurar o registro em log de recursos específicos para o AWS IoT

> **Note**
>
> Também é possível executar esse procedimento com a API usando os métodos na API da AWS que correspondam aos comandos da CLI mostrados aqui.

1. Por padrão, todos os recursos têm nível de log definido como`ERROR`. Para definir os níveis de log padrão ou os níveis de log por tipos de recursos para todos os recursos em sua conta, use oupdate-log-levels-by-resource-typescomando. O exemplo a seguir mostra como criar um arquivo JSON,`Input.jsone` forneça-o como uma entrada para o comando da CLI. Você pode usar esse comando para desabilitar seletivamente o registro em log ou substituir o nível de log padrão para tipos específicos de recursos e eventos.

```
{
    "DefaultLogLevel": "INFO",
    "WirelessDeviceLogOptions":
     [
        {
         "Type": "Sidewalk",
         "LogLevel": "INFO",
         "Events":
          [
            {
              "Event": "Registration",
              "LogLevel": "DISABLED"
            }
          ]
        },
        {
         "Type": "LoRaWAN",
         "LogLevel": "INFO",
         "Events":
          [
            {
             "Event": "Join",
             "LogLevel": "DISABLED"
            },
            {
             "Event": "Rejoin",
             "LogLevel": "ERROR"
            }
          ]
        }
      ],
     "WirelessGatewayLogOptions":
      [
        {
         "Type": "LoRaWAN",
         "LogLevel": "INFO",
         "Events":
          [
            {
             "Event": "CUPS_Request",
             "LogLevel": "DISABLED"
            },
            {
              "Event": "Certificate",
              "LogLevel": "ERROR"
```

```
            }
          ]
        }
      ]
}
```

em que:

WirelessDeviceLogOptions

A lista de opções de registro para um dispositivo sem fio. Cada opção de registro inclui o tipo de dispositivo sem fio (Sidewalk ou LoRawan) e uma lista de opções de registro de eventos de dispositivo sem fio. Cada opção de registro de eventos de dispositivo sem fio pode, opcionalmente, incluir o tipo de evento e seu nível de log.

WirelessGatewayLogOptions

A lista de opções de registro para um gateway sem fio. Cada opção de log inclui o tipo de gateway sem fio (LoRaWAN) e uma lista de opções de log de eventos de gateway sem fio. Cada opção de log de eventos do gateway sem fio pode, opcionalmente, incluir o tipo de evento e seu nível de log.

DefaultLogLevel

O nível de log a ser usado para todos os recursos do. Os valores válidos são: `ERROR`, `INFO` e `DISABLED`. O valor padrão é `INFO`.

LogLevel

O nível de log que você deseja usar para tipos de recursos e eventos individuais. Esses níveis de log substituem o nível de log padrão, como o nível de log`INFO`para o gateway LoRaWAN e níveis de log`DISABLED`e`ERROR`para os dois tipos de evento.

Execute o comando a seguir para fornecer `oInput.json`Arquivo como entrada para o comando. Esse comando não produz nenhuma saída.

```
aws iotwireless update-log-levels-by-resource-types \
    --cli-input-json file://input.json
```

Se você quiser remover as opções de log para dispositivos sem fio e gateways sem fio, execute o seguinte comando.

```
{
    "DefaultLogLevel":"DISABLED",
    "WirelessDeviceLogOptions": [],
    "WirelessGatewayLogOptions":[]
}
```

2. O comando update-log-levels-by-resource-types não retorna nenhuma saída. Usar aget-log-levels-by-resource-typescomando para recuperar informações de registro específicas do recurso. O comando retorna o nível de log padrão e as opções de log do dispositivo sem fio e do gateway sem fio.

   Note

   Oget-log-levels-by-resource-typescomando não pode recuperar diretamente os níveis de log na CloudWatch console do . Você pode usar oget-log-levels-by-resource-typescomando para obter as informações mais recentes em nível de log que você especificou para seus recursos usando oupdate-log-levels-by-resource-typescomando.

```
aws iotwireless get-log-levels-by-resource-types
```

Quando você executa o seguinte comando, ele retorna as informações de registro mais recentes que você especificou comupdate-log-levels-by-resource-types. Por exemplo, se você remover as opções de registro do dispositivo sem fio, execute oget-log-levels-by-resource-typesretornará esse valor como`null`.

```
{
    "DefaultLogLevel": "INFO",
    "WirelessDeviceLogOptions": null,
     "WirelessGatewayLogOptions":
      [
        {
         "Type": "LoRaWAN",
         "LogLevel": "INFO",
         "Events":
          [
            {
             "Event": "CUPS_Request",
             "LogLevel": "DISABLED"
            },
            {
              "Event": "Certificate",
              "LogLevel": "ERROR"
            }
          ]
        }
      ]
}
```

3. Para controlar os níveis de log para gateways sem fio individuais ou recursos de dispositivos sem fio, use os seguintes comandos da CLI:

   - put-resource-log-level
   - get-resource-log-level
   - reset-resource-log-level

   Para obter um exemplo de quando usar essas CLIs, digamos que você tenha um grande número de dispositivos sem fio ou gateways em sua conta que estão sendo registrados. Se você quiser solucionar erros para apenas alguns de seus dispositivos sem fio, você pode desativar o registro de todos os dispositivos sem fio definindo o`DefaultLogLevel`para`DISABLED`, e use oput-resource-log-levelpara definir o`LogLevel`para`ERROR`Para apenas esses dispositivos em sua conta.

```
aws iotwireless put-resource-log-level \
    --resource-identifier
    --resource-type WirelessDevice
    --log-level ERROR
```

   Neste exemplo, o comando define o nível de log como`ERROR`somente para o recurso de dispositivo sem fio especificado e os registros de todos os outros recursos estão desativados. Esse comando não produz nenhuma saída. Para recuperar essas informações e verificar se os níveis de log foram definidos, use oget-resource-log-levelcomando.

4. Na etapa anterior, depois de depurar o problema e resolver o erro, você pode executar oreset-resource-log-levelcomando para redefinir o nível de log para esse recurso`null`. Se você usou o`put-resource-log-level`para definir a substituição no nível de log para mais de um dispositivo sem fio ou recurso de gateway, como para solucionar erros para vários dispositivos, você pode redefinir as

substituições no nível de log de volta para`null`para todos esses recursos usando oreset-all-resource-log-levelscomando.

```
aws iotwireless reset-all-resource-log-levels
```

Esse comando não produz nenhuma saída. Para recuperar as informações de registro para os recursos, execute aget-resource-log-levelcomando.

Próximas etapas

Você aprendeu a criar a função de registro em log e usar oAWS IoT WirelessAPI para configurar o registro em log para oAWS IoT Wirelessrecursos da AWS. Em seguida, para saber mais sobre como monitorar suas entradas de registro, acesseMonitorAWS IoT Wirelessusando CloudWatch Logs (p. 1237).

MonitorAWS IoT Wirelessusando CloudWatch Logs

AWS IoT Wirelesstem mais de 50 CloudWatch As entradas de log que estão habilitadas por padrão. Cada entrada de log descreve o tipo de evento, o nível de log e o tipo de recurso. Para obter mais informações, consulteAWS IoT WirelessRecursos e níveis de log (p. 1229)

Como monitorar seuAWS IoT Wirelessrecursos

Quando o registro estiver habilitado paraAWS IoT Wireless,AWS IoT WirelessA envia eventos de progresso sobre cada mensagem ao passar de seus dispositivos atéAWS IoTe voltar. Por padrão,AWS IoT Wirelessas entradas de registro têm um nível de erro de log padrão. Quando você habilita o registro em log conforme descrito emCriar uma política e uma função de registro em log paraAWS IoT Wireless (p. 1227), você verá mensagens no CloudWatch console que tem um nível de log padrão de`ERROR`. Ao usar esse nível de log, as mensagens mostrarão apenas informações de erro para todos os dispositivos sem fio e recursos de gateway que você estiver usando.

Se você quiser que os logs mostrem informações adicionais, como aquelas que têm um nível de log de`INFO`, ou desabilite registros de alguns de seus dispositivos e mostre mensagens de registro para apenas alguns de seus dispositivos, você pode usar oAWS IoT WirelessAPI de registro em log. Para obter mais informações, consulteConfigurar níveis de log de recursos usando a CLI (p. 1233)

Você também pode criar expressões de filtro para exibir somente as mensagens necessárias.

Antes que você possa visualizarAWS IoT WirelessLogs no console do

Para criar o/aws/iotwirelessO grupo de logs aparece no console do CloudWatch, você deve ter feito o seguinte.

- Habilitar o loginAWS IoT Wireless. Para obter mais informações sobre como habilitar o registro em log noAWS IoT Wireless, consulteConfigurar o registro em log para o AWS IoT Wireless (p. 1227).
- Escreveu algumas entradas de registro executandoAWS IoT Wirelessoperações.

Para criar e usar expressões de filtro de forma mais eficaz, recomendamos que você tente usar CloudWatch Insights conforme descrito nos tópicos a seguir. Também recomendamos que você siga os tópicos na ordem em que eles são apresentados aqui. Isso ajudará você a usar o CloudWatchGrupos de logsprimeiro para aprender sobre os diferentes tipos de recursos, seus tipos de eventos e níveis de log que você pode usar para exibir entradas de registro no console. Você pode aprender a criar expressões de filtro usando o CloudWatch Insights para obter informações mais úteis de seus recursos.

Tópicos

- Exibir CloudWatch AWS IoT WirelessEntradas de log do do (p. 1238)
- Usar o CloudWatch Insights para filtrar logs paraAWS IoT Wireless (p. 1243)

Exibir CloudWatch AWS IoT WirelessEntradas de log do do

Depois de configurar o registro em log paraAWS IoT WirelessConforme descrito emCriar uma política e uma função de registro em log paraAWS IoT Wireless (p. 1227)e gravou algumas entradas de registro, você pode visualizar as entradas de registro no CloudWatch Siga estas etapas.

Visualizar oAWS IoTlogs no CloudWatch Console de grupos de log

NoConsole do CloudWatch, CloudWatch Os logs do são exibidos em um grupo de logs chamado/aws/iotwireless. Para obter mais informações sobre CloudWatch Registros, consulteCloudWatch Logs.

Para visualizar osAWS IoTlogs no CloudWatch console

Navegue até o .Console do CloudWatche escolhaGrupos de logsNo painel de navegação.

1. NoFiltrocaixa de texto, insira**`/aws/iotwireless`**e, em seguida, escolha o`/aws/iotwireless`Grupo de logs).
2. Para ver uma lista completa doAWS IoT Wirelesslogs gerados para sua conta, escolhaPesquisar todos. Para examinar um fluxo de log individual, escolha o ícone de expansão.
3. Para filtrar os fluxos de log, você também pode inserir uma consulta noFiltrar eventosCaixa de texto. Veja algumas consultas para testar:

   - `{ $.logLevel = "ERROR" }`

     Use esse filtro para encontrar todos os logs que têm um nível de log de`ERROR`e você pode expandir os fluxos de erro individuais para ler as mensagens de erro, o que ajudará você a resolvê-las.
   - `{ $.resource = "WirelessGateway" }`

     Encontre todos os registros para o`WirelessGateway`recurso, independentemente do nível de log.
   - `{ $.event = "CUPS_Request" && $.logLevel = "ERROR" }`

     Encontre todos os logs que têm um tipo de evento de`CUPS_Request`e um nível de log de`ERROR`.

Eventos e tipos de recursos

A tabela a seguir mostra os diferentes tipos de eventos para os quais você verá entradas de registro. Os tipos de evento também dependem se o tipo de recurso é um dispositivo sem fio ou um gateway sem fio. Você pode usar o nível de log padrão para os recursos e tipos de evento ou substituir o nível de log padrão especificando um nível de log para cada um deles.

Tipos de eventos com base em recursos usados

Recurso	Tipo de recurso	Tipo de evento	
Gateway wireless	LoRaWAN	• Cups_request • Certificado	

Recurso	Tipo de recurso	Tipo de evento	
Dispositivo sem fio	LoRaWAN	• Ingressar • Junte-se novamente • Uplink_Data • Downlink_Data	
Dispositivo sem fio	Calçada	• Registro • Uplink_Data • Downlink_Data	

O tópico a seguir contém mais informações sobre esses tipos de eventos e as entradas de registro para gateways sem fio e dispositivos sem fio.

Tópicos

- Entradas de registro para gateways sem fio e recursos de dispositivos sem fio (p. 1239)

Entradas de registro para gateways sem fio e recursos de dispositivos sem fio

Depois de habilitar o registro em log, você pode visualizar as entradas de registro para seus gateways sem fio e dispositivos sem fio. A seção a seguir descreve os vários tipos de entradas de registro com base nos tipos de recursos e eventos.

Entradas de log do gateway sem

Esta seção mostra algumas das entradas de registro de exemplo para seus recursos de gateway sem fio que você verá naConsole do CloudWatch. Essas mensagens de registro podem ter o tipo de evento como`CUPS_Request`ou`Certificate`e pode ser configurado para exibir um nível de log de`INFO`,`ERROR`, ou`DISABLED`no nível do recurso ou no nível do evento. Se você quiser ver apenas informações de erro, defina o nível de log como`ERROR`. A mensagem no`ERROR`a entrada de log conterá informações sobre por que ela falhou.

As entradas de registro do recurso de gateway sem fio podem ser classificadas com base nos seguintes tipos de evento:

- Cups_request

  O LoRa Basics Station em execução no gateway envia periodicamente uma solicitação ao Servidor de Configuração e Atualização (CUPS) para atualizações. Para esse tipo de evento, se você definir o nível de log como`INFO`ao configurar a CLI para o recurso de gateway sem fio e, em seguida, nos logs:

  - Se o evento for bem-sucedido, você verá as mensagens de log que têm um`logLevel`do`INFO`. As mensagens incluirão detalhes sobre a resposta do CUPS enviada ao gateway e os detalhes do gateway. A seguir, mostra um exemplo dessa entrada de log. Para obter mais informações sobre `logLevel`e outros campos na entrada de log, consulteAWS IoT WirelessRecursos e níveis de log (p. 1229).

```
{
    "timestamp": "2021-05-13T16:56:08.853Z",
    "resource": "WirelessGateway",
    "wirelessGatewayId": "5da85cc8-3361-4c79-8be3-3360fb87abda",
    "wirelessGatewayType": "LoRaWAN",
    "gatewayEui": "feffff00000000e2",
    "event": "CUPS_Request",
```

```
    "logLevel": "INFO",
    "message": "Sending CUPS response of total length 3213 to GatewayEui:
 feffff00000000e2 with  TC Credentials,"
}
```

- Se houver um erro, você verá entradas de registro que têm um`logLevel`do`ERROR`E as mensagens incluirão detalhes sobre o erro. Exemplos de quando um erro pode ocorrer para o`CUPS_Request`evento incluem: CUPS CRC ausente, incompatibilidade no TC Uri do gateway comAWS IoT Wireless, ausente`IoTWirelessGatewayCertManagerRole`, ou não é possível obter registro de gateway sem fio. O exemplo a seguir mostra uma entrada de log CRC ausente. Para resolver o erro, verifique a configuração do gateway para verificar se você inseriu o CUPS CRC correto.

```
{
    "timestamp": "2021-05-13T16:56:08.853Z",
    "resource": "WirelessGateway",
    "wirelessGatewayId": "5da85cc8-3361-4c79-8be3-3360fb87abda",
    "wirelessGatewayType": "LoRaWAN",
    "gatewayEui": "feffff00000000e2",
    "event": "CUPS_Request",
    "logLevel": "ERROR",
    "message": "The CUPS CRC is missing from the request. Check your gateway setup and
 enter the CUPS CRC,"
}
```

- Certificado

  Essas entradas de registro ajudarão você a verificar se o gateway sem fio apresentou o certificado correto para autenticar a conexão comAWS IoT. Para esse tipo de evento, se você definir o nível de log como`INFO`ao configurar a CLI para o recurso de gateway sem fio e, em seguida, nos logs:

  - Se o evento for bem-sucedido, você verá as mensagens de log que têm um`logLevel`do`INFO`. As mensagens incluirão detalhes sobre o ID do certificado e o identificador do gateway sem fio. A seguir, mostra um exemplo dessa entrada de log. Para obter mais informações sobre o`logLevel`e outros campos na entrada de log, consulteAWS IoT WirelessRecursos e níveis de log (p. 1229).

```
{
    "resource": "WirelessGateway",
    "wirelessGatewayId": "5da85cc8-3361-4c79-8be3-3360fb87abda",
    "wirelessGatewayType": "LoRaWAN",
    "event": "Certificate",
    "logLevel": "INFO",
    "message": "Gateway connection authenticated.
    (CertificateId: b5942a7aee973eda24314e416889227a5e0aa5ed87e6eb89239a83f515dea17c,
 WirelessGatewayId: 5da85cc8-3361-4c79-8be3-3360fb87abda)"
}
```

  - Se houver um erro, você verá entradas de registro que têm um`logLevel`do`ERROR`E as mensagens incluirão detalhes sobre o erro. Exemplos de quando um erro pode ocorrer para o`Certificate`incluem uma ID de certificado inválida, identificador de gateway sem fio ou uma incompatibilidade entre o identificador do gateway sem fio e o ID do certificado. O exemplo a seguir mostra um`ERROR`devido ao identificador de gateway sem fio inválido. Para resolver o erro, verifique os identificadores de gateway.

```
{
    "resource": "WirelessGateway",
    "wirelessGatewayId": "5da85cc8-3361-4c79-8be3-3360fb87abda",
    "wirelessGatewayType": "LoRaWAN",
    "event": "Certificate",
    "logLevel": "INFO",
    "message": "The gateway connection couldn't be authenticated because a provisioned
 gateway associated with the certificate couldn't be found.
```

```
    (CertificateId: 729828e264810f6fc7134daf68056e8fd848afc32bfe8082beeb44116d709d9e)"
}
```

Entradas de log de dispositivos sem

Esta seção mostra algumas das entradas de registro de exemplo para os recursos do dispositivo sem fio que você verá naConsole do CloudWatch. O tipo de evento dessas mensagens de registro depende se você está usando um LoRaWAN ou um dispositivo Sidewalk. Cada recurso de dispositivo sem fio ou tipo de evento pode ser configurado para exibir um nível de log de`INFO`,`ERROR`, ou`DISABLED`.

> **Note**
>
> Sua solicitação não deve conter metadados sem fio LoraWAN e Sidewalk ao mesmo tempo. Para evitar um`ERROR`entrada de registro para este cenário, especifique dados sem fio LoraWAN ou Sidewalk.

Entradas de log de dispositivos LoRawan

As entradas de registro do seu dispositivo sem fio LoRaWAN podem ser classificadas com base nos seguintes tipos de evento:

- **`Join`** e **`Rejoin`**

  Quando você adiciona um dispositivo LoRaWAN e o conecta aAWS IoT Wireless, antes que seu dispositivo possa enviar dados de uplink, você deve concluir um processo chamado`activation`ou`join procedure`. Para obter mais informações, consulteAdicione seu dispositivo sem fio aoAWS IoT Core for LoRaWAN (p. 1140)

  Para esse tipo de evento, se você definir o nível de log como`INFO`ao configurar a CLI para o recurso de gateway sem fio e, em seguida, nos logs:

  - Se o evento for bem-sucedido, você verá as mensagens de log que têm um`logLevel`do`INFO`. As mensagens incluirão detalhes sobre o status de sua solicitação de ingresso ou de volta. A seguir, mostra um exemplo dessa entrada de log. Para obter mais informações sobre o`logLevel`e outros campos na entrada de log, consulteAWS IoT WirelessRecursos e níveis de log (p. 1229).

    ```
    {
        "timestamp": "2021-05-13T16:56:08.853Z",
        "resource": "WirelessDevice",
        "wirelessDeviceType": "LoRaWAN",
        "WirelessDeviceId": "5da85cc8-3361-4c79-8be3-3360fb87abda",
        "devEui": "feffff00000000e2",
        "event": "Rejoin",
        "logLevel": "INFO",
        "message": "Rejoin succeeded"
    }
    ```

  - Se houver um erro, você verá entradas de registro que têm um`logLevel`do`ERROR`E as mensagens incluirão detalhes sobre o erro. Exemplos de quando um erro pode ocorrer para o`Join`e`Rejoin`eventos incluem configuração de região LoRaWAN inválida ou verificação inválida do Código de Integridade de Mensagem (MIC). O exemplo a seguir mostra um erro de junção devido à verificação do MIC. Para resolver o erro, verifique se você inseriu as chaves raiz corretas.

    ```
    {
        "timestamp": "2020-11-24T01:46:50.883481989Z",
        "resource": "WirelessDevice",
        "wirelessDeviceType": "LoRaWAN",
        "WirelessDeviceId": "cb4c087c-1be5-4990-8654-ccf543ee9fff",
        "devEui": "58a0cb000020255c",
        "event": "Join",
    ```

```
    "logLevel": "ERROR",
    "message": "invalid MIC. It's most likely caused by wrong root keys."
}
```

- Uplink_Data e Downlink_Data

  O tipo de evento`Uplink_Data`é usado para mensagens geradas porAWS IoT Wirelessquando a carga útil é enviada do dispositivo sem fio paraAWS IoT. O tipo de evento`Downlink_Data`é usado para mensagens relacionadas a mensagens de downlink enviadas deAWS IoTPara o dispositivo sem fio.

  > **Note**
  >
  > Eventos`Uplink_Data`e`Downlink_Data`aplicam-se aos dispositivos LoRawan e Sidewalk.

  Para esse tipo de evento, se você definir o nível de log como`INFO`ao configurar a CLI para seus dispositivos sem fio e, em seguida, nos registros, você verá:

  - Se o evento for bem-sucedido, você verá as mensagens de log que têm um`logLevel`do`INFO`. As mensagens incluirão detalhes sobre o status da mensagem de uplink ou downlink que foi enviada e o identificador do dispositivo sem fio. A seguir, mostra um exemplo dessa entrada de registro para um dispositivo Sidewalk. Para obter mais informações sobre o`logLevel`e outros campos na entrada de log, consulteAWS IoT WirelessRecursos e níveis de log (p. 1229).

```
{
    "resource": "WirelessDevice",
    "wirelessDeviceId": "5371db88-d63d-481a-868a-e54b6431845d",
    "wirelessDeviceType": "Sidewalk",
    "event": "Downlink_Data",
    "logLevel": "INFO",
    "messageId": "8da04fa8-037d-4ae9-bf67-35c4bb33da71",
    "message": "Message delivery succeeded.  MessageId: 8da04fa8-037d-4ae9-
bf67-35c4bb33da71. AWS IoT Core: {\"message\":\"OK\",\"traceId\":\"038b5b05-a340-
d18a-150d-d5a578233b09\"}"
}
```

  - Se houver um erro, você verá entradas de registro que têm um`logLevel`do`ERROR`, e as mensagens incluirão detalhes sobre o erro, o que o ajudará a resolvê-lo. Exemplos de quando um erro pode ocorrer para o`Registration`event include: problemas de autenticação, solicitações inválidas ou muitas, não é possível criptografar ou descriptografar a carga útil ou não foi possível encontrar o dispositivo sem fio usando o ID especificado. O exemplo a seguir mostra um erro de permissão encontrado ao processar uma mensagem.

```
{
    "resource": "WirelessDevice",
    "wirelessDeviceId": "cb4c087c-1be5-4990-8654-ccf543ee9fff",
    "wirelessDeviceType": "LoRaWAN",
    "event": "Uplink_Data",
    "logLevel": "ERROR",
    "message": "Cannot assume role MessageId: ef38877f-3454-4c99-96ed-5088c1cd8dee.
    Access denied: User: arn:aws:sts::005196538709:assumed-role/
DataRoutingServiceRole/6368b35fd48c445c9a14781b5d5890ed is not authorized to perform:
 sts:AssumeRole on resource: arn:aws:iam::400232685877:role/ExecuteRules_Role\tstatus
 code: 403, request id: 471c3e35-f8f3-4e94-b734-c862f63f4edb"
}
```

Entradas de log do dispositivo de calçada

As entradas de registro do seu dispositivo Sidewalk podem ser classificadas com base nos seguintes tipos de evento:

- **`Registration`**

Essas entradas de registro ajudarão você a monitorar o status de qualquer dispositivo Sidewalk com o qual você está registrandoAWS IoT Wireless. Para esse tipo de evento, se você definir o nível de log como`INFO`ao configurar a CLI para o recurso do dispositivo sem fio e, em seguida, nos registros, você verá mensagens de registro com um`logLevel`do`INFO`e`ERROR`. As mensagens incluirão detalhes sobre o progresso do registro do início até a conclusão.`ERROR`mensagens de registro conterão informações sobre como solucionar problemas com o registro do dispositivo.

A seguir mostra um exemplo para uma mensagem de log com nível de log de`INFO`. Para obter mais informações sobre `o`log`Level`e outros campos na entrada de log, consulteAWS IoT WirelessRecursos e níveis de log (p. 1229).

```
{
    "resource": "WirelessDevice",
    "wirelessDeviceId": "8d0b2775-e19b-4b2a-a351-cb8a2734a504",
    "wirelessDeviceType": "Sidewalk",
    "event": "Registration",
    "logLevel": "INFO",
    "message": "Successfully completed device registration. Amazon SidewalkId =
 2000000002"
}
```

- Uplink_Data e Downlink_Data

  Os tipos de evento`Uplink_Data`e`Downlink_Data`para dispositivos Sidewalk são semelhantes aos tipos de eventos correspondentes para dispositivos LoRaWAN. Para obter mais informações, consulte oUplink_Data e Downlink_Dataseção descrita anteriormente para entradas de registro de dispositivos LoRaWAN.

Next steps (Próximas etapas)

Você aprendeu a exibir entradas de registro para seus recursos e as diferentes entradas de registro que você pode visualizar na CloudWatch console após habilitar o registro em log paraAWS IoT Wireless. Enquanto você pode criar fluxos de filtro usandoGrupos de logs, recomendamos o uso de CloudWatch Insights para criar e usar fluxos de filtro. Para obter mais informações, consulteUsar o CloudWatch Insights para filtrar logs paraAWS IoT Wireless (p. 1243)

Usar o CloudWatch Insights para filtrar logs paraAWS IoT Wireless

Enquanto você pode usar CloudWatch Para criar expressões de filtro, recomendamos que você use CloudWatch insights para criar e usar expressões de filtro de forma mais eficaz, dependendo do aplicativo.

Recomendamos usar primeiro o CloudWatch Grupos de logspara saber mais sobre os diferentes tipos de recursos, seus tipos de eventos e níveis de log que você pode usar para exibir entradas de registro no console. É possível usar os exemplos de algumas expressões de filtro nesta página como referência para criar os seus próprios filtros para oAWS IoT Wirelessrecursos da AWS.

Visualizar oAWS IoTlogs no CloudWatch Console de insights de registros

NoConsole do CloudWatch, CloudWatch Os logs do são exibidos em um grupo de logs chamado/aws/iotwireless. Para obter mais informações sobre CloudWatch Registros, consulteCloudWatch Logs.

Para visualizar osAWS IoTlogs no CloudWatch console

Navegue até o .Console do CloudWatche escolhaInsights de registrosNo painel de navegação.

1. NoFiltrocaixa de texto, insira**`/aws/iotwireless`**e, em seguida, escolha o`/aws/iotwireless`Insights de registros.
2. Para ver uma lista completa de grupos de logs, escolhaSelecionar grupo (s) de log). Para examinar grupos de log paraAWS IoT Wireless, escolha`/aws/iotwireless`.

Agora você pode começar a inserir consultas para filtrar os grupos de log. As seções a seguir contêm algumas consultas úteis que ajudarão você a obter insights sobre suas métricas de recursos.

Crie consultas úteis para filtrar e obter insights paraAWS IoT Wireless

Você pode usar expressões de filtro para mostrar informações de log úteis adicionais com CloudWatch Insights. A seguir, mostra algumas consultas de amostra:

Mostrar somente registros para tipos de recursos específicos

Você pode criar uma consulta que o ajudará a mostrar logs apenas para tipos de recursos específicos, como um gateway LoRaWAN ou um dispositivo Sidewalk. Por exemplo, para filtrar logs para mostrar somente mensagens para dispositivos Sidewalk, você pode inserir a seguinte consulta e escolherExecutar consulta. Para salvar essa consulta, escolha Salvar.

```
fields @message
| filter @message like /Sidewalk/
```

Depois que a consulta for executada, você verá os resultados naLogs, que mostra os carimbos de data/hora para registros relacionados aos dispositivos Sidewalk em sua conta. Você também verá um gráfico de barras, que mostrará a hora em que os eventos ocorreram, se houver eventos que ocorreram anteriormente relacionados ao seu dispositivo Sidewalk. A seguir mostra um exemplo se você expandir um dos resultados naLogsGuia. Como alternativa, se você quiser solucionar erros relacionados aos dispositivos Sidewalk, você pode adicionar outro filtro que defina o nível de log como`ERROR`e mostre apenas informações de erro.

```
Field           Value
@ingestionTime      1623894967640
@log              954314929104:/aws/iotwireless
@logStream    WirelessDevice-
Downlink_Data-715adccfb34170214ec2f6667ddfa13cb5af2c3ddfc52fbeee0e554a2e780bed
@message      {
                    "resource": "WirelessDevice",
                    "wirelessDeviceId": "3b058d05-4e84-4e1a-b026-4932bddf978d",
                    "wirelessDeviceType": "Sidewalk",
                    "devEui": "feffff000000011a",
                    "event": "Downlink_Data",
                    "logLevel": "INFO",
                    "messageId": "7e752a10-28f5-45a5-923f-6fa7133fedda",
                    "message": "Successfully sent downlink message. Amazon SidewalkId =
 2000000006, Sequence number = 0"
                    }
@timestamp          1623894967640
devEui            feffff000000011a
event         Downlink_Data
logLevel            INFO
message           Successfully sent downlink message. Amazon SidewalkId = 2000000006,
 Sequence number = 0
messageId     7e752a10-28f5-45a5-923f-6fa7133fedda
resource      WirelessDevice
wirelessDeviceId    3b058d05-4e84-4e1a-b026-4932bddf978d
wirelessDeviceType  Sidewalk
```

Mostrar mensagens ou eventos específicos

Você pode criar uma consulta que o ajudará a mostrar mensagens específicas e observar quando os eventos ocorreram. Por exemplo, se você quiser ver quando sua mensagem de downlink foi enviada do seu dispositivo sem fio LoraWAN, você pode inserir a seguinte consulta e escolherExecutar consulta. Para salvar essa consulta, escolha Salvar.

```
filter @message like /Downlink message sent/
```

Depois que a consulta for executada, você verá os resultados naLogs, que mostra os carimbos de data/hora quando a mensagem de downlink foi enviada com êxito para o seu dispositivo sem fio. Você também verá um gráfico de barras, que mostrará a hora em que uma mensagem de downlink foi enviada, se houver mensagens de downlink enviadas anteriormente para o seu dispositivo sem fio. A seguir mostra um exemplo se você expandir um dos resultados naLogsGuia. Como alternativa, se uma mensagem de downlink não tiver sido enviada, você poderá modificar a consulta para exibir apenas resultados para quando a mensagem não foi enviada para que você possa depurar o problema.

```
Field           Value
@ingestionTime      1623884043676
@log              954314929104:/aws/iotwireless
@logStream    WirelessDevice-
Downlink_Data-42d0e6d09ba4d7015f4e9756fcdc616d401cd85fe3ac19854d9fbd866153c872
@message      {
                    "timestamp": "2021-06-16T22:54:00.770493863Z",
                    "resource": "WirelessDevice",
                    "wirelessDeviceId": "3b058d05-4e84-4e1a-b026-4932bddf978d",
                    "wirelessDeviceType": "LoRaWAN",
                    "devEui": "feffff000000011a",
                    "event": "Downlink_Data",
                    "logLevel": "INFO",
                    "messageId": "7e752a10-28f5-45a5-923f-6fa7133fedda",
                    "message": "Downlink message sent. MessageId:
 7e752a10-28f5-45a5-923f-6fa7133fedda"
                    }
@timestamp          1623884040858
devEui            feffff000000011a
event         Downlink_Data
logLevel            INFO
message           Downlink message sent. MessageId: 7e752a10-28f5-45a5-923f-6fa7133fedda
messageId     7e752a10-28f5-45a5-923f-6fa7133fedda
resource      WirelessDevice
timestamp     2021-06-16T22:54:00.770493863Z
wirelessDeviceId    3b058d05-4e84-4e1a-b026-4932bddf978d
wirelessDeviceType  LoRaWAN
```

Next steps (Próximas etapas)

Você aprendeu a usar CloudWatch Insights para obter informações mais úteis criando consultas para filtrar mensagens de log. Você pode combinar alguns dos filtros descritos anteriormente e projetar seus próprios filtros dependendo do recurso que você está monitorando. Para obter mais informações sobre como usar o CloudWatch Insights, consulteAnalisar dados de log com CloudWatch Insights do.

Depois de criar consultas com CloudWatch Insights, se você os salvou, poderá carregar e executar as consultas salvas conforme necessário. Como alternativa, se você clicar no botãoHistóricobotão no CloudWatch Insights de registrosconsole, você pode visualizar as consultas executadas anteriormente e executá-las novamente conforme necessário, ou modificá-las ainda mais criando consultas adicionais.

Notificações de eventos paraAWS IoT Wireless

AWS IoT Wirelesspode publicar mensagens para notificá-lo sobre eventos para dispositivos LoRaWAN e Sidewalk que você a bordoAWS IoT Core. Por exemplo, você pode ser notificado de eventos como quando os dispositivos Sidewalk em sua conta foram provisionados ou registrados.

Como seus recursos podem ser notificados sobre eventos

As notificações de eventos são publicadas quando ocorrem determinados eventos. Por exemplo, eventos são gerados quando o dispositivo Sidewalk é provisionado. Cada evento faz com que uma única notificação de evento seja enviada. As notificações de eventos são publicadas por meio do MQTT com uma carga JSON. O conteúdo da carga depende do tipo do evento.

> **Note**
>
> As notificações de eventos são publicadas pelo menos uma vez. É possível que elas sejam publicadas mais de uma vez. A ordenação das notificações de eventos não é garantida.

Eventos e tipos de recursos

A tabela a seguir mostra os diferentes tipos de eventos para os quais você receberá notificações. Os tipos de evento dependem se o tipo de recurso é um dispositivo sem fio, um gateway sem fio ou uma conta do Sidewalk. Você também pode ativar eventos para seus recursos no nível do recurso, que se aplica a todos os recursos de um tipo específico ou para recursos selecionados, conforme descrito na seção a seguir. Para obter mais informações sobre os diferentes tipos de evento, consulteNotificações de eventos para recursos LoraWAN (p. 1252)eNotificações de eventos para recursos Sidewalk (p. 1257).

Tipos de eventos com base em recursos

Recurso	Tipo de recurso	Tipo de evento	
Dispositivo sem fio	LoRaWAN	Ingressar	
	Calçada	• Estado de registro do dispositivo • Proximidade	
Gateway wireless	LoRaWAN	Status de conexão	
Conta de calçada	Calçada	• Estado de registro do dispositivo • Proximidade	

Política para receber notificações de eventos sem fio

Para receber notificações de eventos, seu dispositivo deve usar uma política adequada que permita que ele se conecte aoAWS IoTgateway de dispositivos e assinar tópicos de eventos MQTT. Você também deve assinar os filtros apropriados dos tópicos.

Veja a seguir um exemplo da política necessária para o recebimento de notificações para os vários eventos sem fio.

```
{
    "Version":"2012-10-17",
    "Statement":[{
        "Effect":"Allow",
        "Action":[
            "iot:Subscribe",
            "iot:Receive"
        ],
        "Resource":[
            "arn:aws:iotwireless:region:account:/$aws/iotwireless/events/join/*",
            "arn:aws:iotwireless:region:account:/$aws/iotwireless/events/connection_status/
*"
            "arn:aws:iotwireless:region:account:/$aws/iotwireless/events/
device_registration_state/*",
            "arn:aws:iotwireless:region:account:/$aws/iotwireless/events/proximity/*"
        ]
    }]
}
```

Formato de tópicos MQTT para eventos sem fio

Para enviar notificações de eventos para seus recursos sem fio,AWS IoTUtiliza tópicos reservados MQTT que começam com um cifrão ($). Você pode publicar e assinar esses tópicos reservados. No entanto, não é possível criar tópicos que comecem com um cifrão.

> **Note**
>
> Tópicos do MQTT são específicos para oConta da AWSe use o formato`arn:aws:iotwireless:`*`aws-region:AWS-account-ID`*`:topic/Topic`. Para obter mais informações, consulte Tópicos MQTT (p. 99).

Tópicos MQTT reservados para dispositivos sem fio usam o seguinte formato:

- Tópicos em nível de recurso

  Esses tópicos se aplicam a todos os recursos de um determinado tipo em seuConta da AWSque você já embarcouAWS IoT Wireless.

  ```
  $aws/iotwireless/events/{eventName}/{eventType}/{resourceType}/resources
  ```

- Tópicos no nível do identificador

  Esses tópicos se aplicam a recursos selecionados de um tipo específico em seuConta da AWSque você já embarcouAWS IoT Wireless, especificado pelo identificador de recurso.

  ```
  $aws/iotwireless/events/{eventName}/{eventType}/{resourceType}/
  {resourceIdentifierType}/{resourceID}/{id}
  ```

Para obter mais informações sobre tópicos do nível de recurso e identificador, consulteConfigurações de evento (p. 1249).

A tabela a seguir mostra exemplos de tópicos do MQTT para os vários eventos:

Tópicos MQTT e eventos

Evento	Tópico MQTT	Observações

Estado de registro do dispositivo de calçada	• Tópico em nível de recurso
	
	`$aws/iotwireless/events/device_registration_state/{eventType}/sidewalk/wireless_devices`
	• Tópico no nível do identificador
	
	`$aws/iotwireless/events/device_registration_state/{eventType}/sidewalk/{resourceType}/{resourceID}/{id}`
		• `{eventType}`Pode ser`registered`ou`provisioned`
		• `{resourceType}`Pode ser`sidewalk_accounts`ou`wireless_devices`
		• `{resourceID}`é o`amazon_id`pelo`sidewalk_accounts`e`wireless_device_`

Proximidade na calçada	• Tópico em nível de recurso
	
	`$aws/iotwireless/events/proximity/{eventType}/sidewalk/wireless_devices`
	• Tópico no nível do identificador
	
	`$aws/iotwireless/events/proximity/{eventType}/sidewalk/{resourceType}/{resourceID}/{id}`
		• `{eventType}`Pode ser`beacon_discovered`ou`beacon_lost`
		• `{resourceType}`Pode ser`sidewalk_accounts`ou`wireless_devices`
		• `{resourceID}`é o`amazon_id`pelo`sidewalk_accounts`e`wireless_device_`

LoRaWAN junte-se	• Tópico em nível de recurso
	
	`$aws/iotwireless/events/join/{eventType}/lorawan/wireless_devices`
	• Tópico no nível do identificador
	
	`$aws/iotwireless/events/join/{eventType}/lorawan/wireless_devices/{resourceID}/{id}`
		• `{eventType}`Pode ser`join_req_0_received`ou`join_req_2_received`ou`joi`
		• `{resourceID}`Pode ser`wireless_device_id`ou`dev_eui`

Status de conexão do gateway LoRawan	• Tópico em nível de recurso	• `{eventType}`Pode ser`connected`ou`disconnected`

Evento	Tópico MQTT	Observações

	`$aws/iotwireless/events/join/{eventType}/lorawan/wireless_gateways`
	• Tópico no nível do identificador
	
	`$aws/iotwireless/events/join/{eventType}/lorawan/wireless_gateways/{resourceID}/{id}`
		• `{resourceID}`Pode ser`wireless_gateway_id`ou`gateway_eui`

Para obter mais informações sobre os diferentes eventos, consulteNotificações de eventos para recursos LoraWAN (p. 1252)eNotificações de eventos para recursos Sidewalk (p. 1257).

Se você se inscreveu nesses tópicos, será notificado quando uma mensagem for publicada em um dos tópicos de notificação de eventos. Para obter mais informações, consulte Tópicos reservados (p. 101).

Preços para eventos sem fio

Para obter mais informações sobre a definição de preço do se inscrever em eventos e como receber notificações, consulteAWS IoT Corepreços.

Ativar eventos para recursos sem fio

Antes que os assinantes dos tópicos reservados possam receber mensagens, você deve ativar as notificações de eventos. Para fazer isso, você pode usar oAWS Management Console, ou oAWS IoT WirelessAPI ouAWS CLI.

Configurações de evento

Você pode configurar eventos para enviar notificações para todos os recursos que pertencem a um tipo específico ou para recursos sem fio individuais. O tipo de recurso pode ser um gateway sem fio, uma conta de parceiro Sidewalk ou um dispositivo sem fio, que pode ser um dispositivo LoRaWAN ou Sidewalk. Para obter informações sobre o tipo de eventos que você pode ativar para seus dispositivos sem fio, consulteTipos de eventos para recursos da LoRaWAN (p. 1252)eTipos de eventos para recursos de calçada (p. 1257).

Todos os recursos do

Você pode habilitar eventos de forma que todos os recursos em seuConta da AWSque pertencem a um tipo de recurso específico recebe notificações. Por exemplo, você pode habilitar um evento que o notifica sobre alterações no status da conexão para todos os gateways LoraWAN que você tenha integradoAWS IoT Core for LoRaWAN. Monitorar esses eventos ajudará você a ser notificado em casos como quando certos gateways LoRawan em sua frota de recursos forem desconectados ou se um farol for perdido por vários dispositivos Sidewalk em seuConta da AWS.

Recursos individuais

Você também pode adicionar recursos individuais do LoRaWAN e do Sidewalk à configuração do seu evento e habilitar notificações para eles. Isso ajudará você a monitorar recursos individuais de um tipo específico. Por exemplo, você pode adicionar dispositivos LoRaWAN e Sidewalk selecionados à sua configuração e receber notificações para eventos de estado de registro de ingresso ou dispositivo para esses recursos.

Pré-requisitos

Seu recurso LoRaWAN ou Sidewalk deve ter uma política apropriada que lhe permita receber notificações de eventos. Para obter mais informações, consulte Política para receber notificações de eventos sem fio (p. 1246).

Ativar notificações usando oAWS Management Console

Para habilitar mensagens de eventos do console, acesse oConfigurações Guia doAWS IoTConsole e, em seguida, vá para oNotificação de evento LorAwan e Sidewalkseção.

Você pode ativar notificações para todos os recursos em seuConta da AWSque pertencem a um tipo de recurso específico e os monitoram.

Para habilitar notificações para todos os recursos

1. NoNotificação de evento LorAwan e Sidewalkseção, vá para oTodos os recursosGuia, selecioneAçãoe, depois, escolhaGerenciar eventos.
2. Habilite os eventos que você deseja monitorar e escolhaAtualizar eventos. Se você não quiser mais monitorar determinados eventos, escolhaAçãoe escolhaGerenciar eventose, em seguida, desative esses eventos.

Você também pode ativar notificações para recursos individuais em seuConta da AWSque pertencem a um tipo de recurso específico e os monitoram.

Para habilitar notificações para recursos individuais

1. NoNotificação de evento LorAwan e Sidewalkseção, selecioneAçãoe, depois, escolhaAdicionar recursos.
2. Selecione os recursos e eventos para os quais deseja receber notificações:
   a. Escolha se você deseja monitorar eventos de seuRecursos LoRaWANouRecursos de calçada.
   b. Dependendo do tipo de recurso, é possível escolher os eventos que deseja ativar para os recursos. Em seguida, você pode se inscrever nesses eventos e receber notificações. Se escolher:
      - Recursos LoRaWAN: Você pode ativarunireventos para seus dispositivos LoRaWAN ouStatus de conexãoeventos para seus gateways LoraWAN.
      - Recursos de calçada: Você pode ativarEstado de registro do dispositivoouproximidadeeventos ou ambos para suas contas de parceiros Sidewalk e dispositivos Sidewalk.
3. Dependendo do tipo de recurso e dos eventos que você escolheu, selecione os dispositivos sem fio ou gateways que você deseja monitorar. É possível selecionar até 250 recursos para todos os recursos combinados.
4. SelecioneEnviarpara adicionar seus recursos.

Os recursos que você adiciona aparecerão com os tópicos do MQTT na guia para o tipo de recurso naNotificação de evento LorAwan e SidewalkSeção do console.

- LoRaWAN junte-seeventos e eventos para seus dispositivos Sidewalk aparecerão naDispositivos sem fioSeção do console.
- Status de conexãoeventos para seus gateways LoRawan aparecerão noGateways sem fioseção.
- Estado de registro do dispositivoeproximidadeeventos para suas contas do Sidewalk aparecerão naContas de calçadaGuia.

Assine tópicos usando o cliente MQTT

Dependendo se você ativou eventos para todos os recursos ou para tipos de recursos individuais, os eventos que você ativou aparecerão no console com seus tópicos MQTT noTodos os recursosou a guia para o tipo de recurso especificado.

- Se você escolher um dos tópicos MQTT, é possível ir ao cliente MQTT para se inscrever nesses tópicos e receber mensagens.
- Se você adicionou vários eventos, poderá se inscrever em vários tópicos de eventos e receber notificações para eles. Para assinar vários tópicos, escolha seus tópicos e escolhaAçãoe, depois, escolhaSubscribe.

Ativar notificações usando oAWS CLI

Você pode configurar eventos e adicionar recursos à sua configuração usando oAWS IoT WirelessAPI ou o comandoAWS CLI.

Ativar notificações para todos os recursos

Você pode ativar notificações para todos os recursos em seuConta da AWSque pertencem a um tipo de recurso específico e os monitoram usando oUpdateEventConfigurationByResourceTypesAPI ou o comando`update-event-configuration-by-resource-types`Comando da CLI. Por exemplo:

```
aws iotwireless update-event-configuration-by-resource-types \
   --cli-input-json input.json
```

Conteúdo da input.json

```
{
   "DeviceRegistrationState": {
      "Sidewalk": {
         "AmazonIdEventTopic": "Enabled"
      }
   },
   "ConnectionStatus": {
      "LoRaWAN": {
         "WirelessGatewayEventTopic": "Enabled"
      }
   }
}
```

> **Note**
>
> Todas as aspas (") são recuadas com uma barra invertida (\).

Você pode obter a configuração atual do evento chamando oGetEventConfigurationByResourceTypesAPI ou usando o`get-event-configuration-by-resource-types`Comando da CLI. Por exemplo:

```
aws iotwireless get-event-configuration-by-resource-types
```

Habilitar notificações para recursos individuais

Para adicionar recursos individuais à configuração do evento e controlar quais eventos são publicados usando a API ou CLI, chame oUpdateResourceEventConfigurationAPI ou use o`update-resource-event-configuration`Comando da CLI. Por exemplo:

```
aws iotwireless update-resource-event-configuration \
   --identifer 1ffd32c8-8130-4194-96df-622f072a315f \
   --identifier-type WirelessDeviceId \
   --cli-input-json input.json
```

Conteúdo da input.json

```
{
   "Join": {
      "LoRaWAN": {
         "DevEuiEventTopic": "Disabled"
      },
      "WirelessDeviceIdEventTopic": "Enabled"
   }
}
```

> **Note**
>
> Todas as aspas (") são recuadas com uma barra invertida (\).

Você pode obter a configuração atual do evento chamando oGetResourceEventConfigurationAPI ou usando o`get-resource-event-configuration`Comando da CLI. Por exemplo:

```
aws iotwireless get-resource-event-configuration \
    --identifier-type WirelessDeviceId \
    --identifier 1ffd32c8-8130-4194-96df-622f072a315f
```

Listar configurações de eventos

Você também pode usar oAWS IoT WirelessAPI ou o comandoAWS CLIpara listar configurações de eventos em que pelo menos um tópico de evento foi ativado. Para listar as configurações, use oListEventConfigurationsOperação da API ou usando o`list-event-configurations`Comando da CLI. Por exemplo:

```
aws iotwireless list-event-configurations --resource-type WirelessDevice
```

Notificações de eventos para recursos LoraWAN

Você pode usar oAWS Management ConsoleouAWS IoT WirelessOperações de API para notificá-lo de eventos para seus dispositivos e gateways Lorawan. Para obter mais informações sobre notificações de eventos do e como habilitá-las, consulteNotificações de eventos paraAWS IoT Wireless (p. 1246)eAtivar eventos para recursos sem fio (p. 1249).

Tipos de eventos para recursos da LoRaWAN

Os eventos que você pode ativar para seus recursos do LoRaWAN incluem:

- Participe de eventos que o notificam sobre participar de eventos para o seu dispositivo LoraWAN. Você receberá notificações quando um dispositivo entrar em contato comAWS IoT Core for LoRaWAN, ou quando uma solicitação de rejunção do tipo 0 ou do tipo 2 é recebida.
- Eventos de status de conexão que o notificam quando o status da conexão do gateway LoraWAN muda para conectado ou desconectado.

As seções a seguir contêm mais informações sobre os eventos para seus recursos do LoRaWAN:

Tópicos

- LoRaWAN junte-se a eventos (p. 1253)
- Eventos de status de conexão (p. 1255)

LoRaWAN junte-se a eventos

AWS IoT Core for LoRaWANpode publicar mensagens para notificá-lo sobre eventos de ingresso para dispositivos LoraWAN aos quais você a bordoAWS IoT. Participar de eventos notificá-lo quando uma solicitação de ingresso ou de volta do tipo 0 ou do tipo 2 é recebida e o dispositivo se juntou aoAWS IoT Core for LoRaWAN.

Como funcionam os eventos da junção

Quando você embarca em seus dispositivos LoraWAN comAWS IoT Core for LoRaWAN,AWS IoT Core for LoRaWANExecute umunirprocedimento para seu dispositivo comAWS IoT Core for LoRaWAN. Seu dispositivo é ativado para uso e pode enviar uma mensagem de uplink para indicar que ele está disponível. Depois que o dispositivo se juntar, as mensagens de uplink e downlink podem ser trocadas entre o dispositivo eAWS IoT Core for LoRaWAN. Para obter mais informações sobre como integrar seu dispositivo, consulteIncorporar seus dispositivosAWS IoT Core for LoRaWAN (p. 1139).

Você pode habilitar eventos para notificá-lo quando seu dispositivo se juntar aoAWS IoT Core for LoRaWAN. Você também será notificado se o evento de junção falhar e quando uma solicitação de rejunção do tipo 0 ou do tipo 2 for recebida e quando for aceita.

Ativar eventos de entrada do LorAwan

Antes que os assinantes dos tópicos reservados da LoRaWAN ingressem possam receber mensagens, você deve habilitar notificações de eventos para eles a partir doAWS Management Consoleou usando a API ou a CLI do. Você pode ativar esses eventos para todos os recursos do LoRaWAN em seuConta da AWSou para recursos selecionados. Para obter informações sobre como ativar esses eventos, consulteAtivar eventos para recursos sem fio (p. 1249).

Formato de tópicos MQTT para eventos LoraWAN

Tópicos MQTT reservados para dispositivos LoRaWAN usam o seguinte formato. Se você se inscreveu nesses tópicos, todos os dispositivos LoraWAN registrados no seuConta da AWSpode receber a notificação:

- Tópicos em nível de recurso

  ```
  $aws/iotwireless/events/{eventName}/{eventType}/lorawan/wireless_devices
  ```

- Tópicos identi

  ```
  $aws/iotwireless/events/{eventName}/{eventType}/lorawan/wireless_devices/
  {resourceID}/{id}
  ```

Onde:

{eventName}

{eventName} deve ser`join`.

{eventType}

{eventType} pode ser:

- `join_req_received`
- `rejoin_req_0_received`
- `rejoin_req_2_received`
- `join_accepted`

{resourceId}

{resourceId} pode ser`dev_eui`ou`wireless_device_id`.

Por exemplo, você pode se inscrever nos tópicos a seguir para receber uma notificação de evento quandoAWS IoT Core for LoRaWANaceitou uma solicitação de ingresso de seus dispositivos.

```
$aws/iotwireless/events/join/join_accepted/lorawan/wireless_devices/
wireless_device_id/{id}
```

Você também pode usar o+caractere curinga para assinar vários tópicos ao mesmo tempo. O+Caractere curinga corresponde a qualquer string no nível que contém o caractere, como o tópico a seguir:

```
$aws/iotwireless/events/join/join_req_received/lorawan/wireless_devices/
wireless_device_id/+
```

> **Note**
>
> Você não pode usar o caractere curinga#para se inscrever nos tópicos reservados. Para obter mais informações sobre os filtros de tópicos, consulteFiltros de tópicos (p. 100).

Para obter mais informações sobre como usar o+curinga ao se inscrever em tópicos, consulteFiltros de tópicos (p. 100).

Carga útil da mensagem para o evento de ingresso do LoraWAN

A seguir, mostra a carga útil da mensagem para o evento de ingresso do LoraWAN.

```
{
 // General fields
    "eventId": "string",
    "eventType": "join_req_received|rejoin_req_0_received|rejoin_req_2_received|
join_accepted",
    "WirelessDeviceId": "string",
    "timestamp": "timestamp",

 // Event-specific fields
    "LoRaWAN": {
        "DevEui": "string",

        // The fields below are optional indicating that it can be a null value.
        "DevAddr": "string",
        "JoinEui": "string",
        "AppEui": "string",
    }
```

```
}
```

A carga contém os seguintes atributos:

eventId

Um ID de evento exclusivo gerado porAWS IoT Core for LoRaWAN(string).

eventType

O tipo de evento que ocorreu. Pode ser um dos valores a seguir:

- `join_req_received`: Este campo mostrará os parâmetros do EUI`JoinEui`ou`AppEui`
- `rejoin_req_0_received`
- `rejoin_req_2_received`
- `join_accepted`: Esse campo mostrará o`NetId`e`DevAddr`.

WirelessDeviceID

O ID do dispositivo LoRawan.

timestamp

A data e hora do Unix de quando o evento ocorreu.

DevEui

O identificador exclusivo do dispositivo encontrado na etiqueta do dispositivo ou na documentação do dispositivo.

DevAddr e EUIs (opcional)

Esses campos são o endereço do dispositivo opcional e os parâmetros EUI`JoinEUI`ou`AppEUI`.

Eventos de status de conexão

AWS IoT Core for LoRaWANpode publicar mensagens para notificá-lo sobre eventos de status de conexão para gateways LoRaWAN que você a bordoAWS IoT. Os eventos de status da conexão notificam você quando o status da conexão de um gateway LoRaWAN muda para conectado ou desconectado.

Como os eventos de status da conexão funcionam

Depois de integrar sua porta de entrada paraAWS IoT Core for LoRaWANVocê pode conectar o gateway aoAWS IoT Core for LoRaWANE verifique o status da conexão. Este evento o notifica quando o status da conexão do gateway muda para conectado ou desconectado. Para obter mais informações sobre integração e conexão de seu gateway aoAWS IoT Core for LoRaWAN, consulteIncorporar seus gateways paraAWS IoT Core for LoRaWAN (p. 1133)eConnect seu gateway LoRaWAN e verifique o status da conexão (p. 1137).

Formato de tópicos MQTT para gateways LoRaWAN

Tópicos MQTT reservados para gateways LoRaWAN usam o seguinte formato. Se você se inscreveu nesses tópicos, todos os gateways LoraWAN registrados no seuConta da AWSpode receber a notificação:

- Para tópicos em nível de recurso:

  ```
  $aws/iotwireless/events/{eventName}/{eventType}/lorawan/wireless_gateways
  ```

- Para tópicos de identificador:

```
$aws/iotwireless/events/{eventName}/{eventType}/lorawan/wireless_gateways/
{resourceID}/{id}
```

Onde:

{eventName}

{eventName} deve ser`connection_status`.

{eventType}

{eventType} pode ser`connected`ou`disconnected`.

{resourceId}

{resourceId} pode ser`gateway_eui`ou`wireless_gateway_id`.

Por exemplo, você pode se inscrever nos tópicos a seguir para receber uma notificação de evento quando todos os seus gateways estiverem conectados aoAWS IoT Core for LoRaWAN:

```
$aws/iotwireless/events/connection_status/connected/lorawan/wireless_gateways/
wireless_gateway_id/{id}
```

Você também pode usar o+caractere curinga para assinar vários tópicos ao mesmo tempo. O+Caractere curinga corresponde a qualquer string no nível que contém o caractere, como o tópico a seguir:

```
$aws/iotwireless/events/connection_status/connected/lorawan/wireless_gateways/
wireless_gateway_id/+
```

Note

Você não pode usar o caractere curinga#para se inscrever nos tópicos reservados. Para obter mais informações sobre os filtros de tópicos, consulteFiltros de tópicos (p. 100).

Para obter mais informações sobre como usar o+curinga ao se inscrever em tópicos, consulteFiltros de tópicos (p. 100).

Carga útil da mensagem para eventos de status de conexão

A seguir, mostra a carga útil da mensagem para o evento de status da conexão.

```
{
 // General fields
    "eventId": "string",
    "eventType": "connected|disconnected",
    "WirelessGatewayId": "string",
    "timestamp": "timestamp",

 // Event-specific fields
    "LoRaWAN": {
        "GatewayEui": "string"
    }
}
```

A carga contém os seguintes atributos:

eventId

Um ID de evento exclusivo gerado porAWS IoT Core for LoRaWAN(string).

eventType

O tipo de evento que ocorreu. Pode ser `connected` ou `disconnected`.

WirelessGatewayID

O ID do gateway LoRawan.

timestamp

A data e hora do Unix de quando o evento ocorreu.

GatewayEui

O identificador exclusivo do gateway encontrado no rótulo do gateway ou na documentação do gateway.

Notificações de eventos para recursos Sidewalk

Você pode usar oAWS Management ConsoleouAWS IoT WirelessOperações de API para notificá-lo sobre eventos de seus dispositivos Sidewalk e contas de parceiros. Para obter mais informações sobre notificações de eventos do e como habilitá-las, consulteNotificações de eventos paraAWS IoT Wireless (p. 1246)eAtivar eventos para recursos sem fio (p. 1249).

Tipos de eventos para recursos de calçada

Os eventos que você pode ativar para seus recursos do Sidewalk incluem:

- Eventos de dispositivo que notificam você sobre alterações no estado do seu dispositivo Sidewalk, como quando o dispositivo foi registrado e está pronto para uso.
- Eventos de proximidade que notificam você quandoAWS IoT Wirelessrecebe uma notificação do Amazon Sidewalk de que um beacon foi descoberto ou perdido.

As seções a seguir contêm mais informações sobre os eventos para seus recursos do Sidewalk:

Tópicos

- Eventos de estado de registro de dispositivos (p. 1257)
- Eventos de proximidade (p. 1259)

Eventos de estado de registro de dispositivos

Eventos de estado de registro do dispositivo publicam notificações de eventos quando há uma alteração no estado de registro do dispositivo, como quando um dispositivo Sidewalk foi provisionado ou registrado. Os eventos fornecem informações sobre os diferentes estados pelos quais o dispositivo passa de quando é provisionado até quando foi registrado.

Como funcionam os eventos de estado de registro de dispositivos

Quando você integra seu dispositivo Sidewalk com o Amazon Sidewalk eAWS IoT Wireless,AWS IoT WirelessExecute um`create`operação e adiciona seu dispositivo Sidewalk ao seuConta da AWS. Em seguida, o dispositivo entra no estado provisionado e o`eventType`torna-se`provisioned`. Para obter mais informações sobre como integrar seu dispositivo, consulteDispositivos Sidewalk integrados com Amazon Sidewalk Integration paraAWS IoT Core (p. 1219).

Depois que o dispositivo tiver sido`provisioned`, o Amazon Sidewalk executa um`register`operação para registrar seu dispositivo Sidewalk comAWS IoT Wireless. O processo de registro é iniciado, onde as chaves de criptografia e sessão são configuradas comAWS IoT. Quando o dispositivo é registrado, o`eventType`torna-se`registered`E seu dispositivo está pronto para uso.

Depois que o dispositivo tiver sido`registered`, Sidewalk pode enviar um pedido para`deregister`Seu dispositivo.AWS IoT Wirelessem seguida, atende à solicitação e altera o estado do dispositivo de volta para`provisioned`. Para obter mais informações sobre os estados do dispositivo, consulteDisviceState.

Ativar notificações para eventos de estado de registro de

Antes que os assinantes do estado de registro do dispositivo tópicos reservados possam receber mensagens, você deve habilitar notificações de eventos para eles a partir doAWS Management Consoleou usando a API ou a CLI do. Você pode ativar esses eventos para todos os recursos do Sidewalk em seuConta da AWSou para recursos selecionados. Para obter informações sobre como ativar esses eventos, consulteAtivar eventos para recursos sem fio (p. 1249).

Formato de tópicos MQTT para eventos de estado de registro de dispositivos

Para notificá-lo sobre eventos estaduais de registro do dispositivo, você pode assinar tópicos reservados do MQTT que começam com um sinal de dólar ($). Para obter mais informações, consulte Tópicos MQTT (p. 99).

Tópicos MQTT reservados para eventos de estado de registro de dispositivos Sidewalk usam o seguinte formato:

- Para tópicos em nível de recurso:

  ```
  $aws/iotwireless/events/{eventName}/{eventType}/sidewalk/wireless_devices
  ```

- Para tópicos de identificador:

  ```
  $aws/iotwireless/events/{eventName}/{eventType}/sidewalk/{resourceType}/
  {resourceID}/{id}
  ```

Onde:

{eventName}

{eventName} deve ser`device_registation_state`.

{eventType}

{eventType} pode ser`provisioned`ou`registered`.

{resourceType}

{resourceType} pode ser`sidewalk_accounts`ou`wireless_devices`.

{resourceId}

{resourceId} é`amazon_id`para {resourceType} de`sidewalk_accounts`e`wireless_device_id`para {resourceType} de`wireless_devices`.

Você também pode usar o+caractere curinga para assinar vários tópicos ao mesmo tempo. O+Caractere curinga corresponde a qualquer string no nível que contém o caractere. Por exemplo, caso você queira ser notificado de todos os tipos de evento possíveis (`provisioned`e`deregistered`) e para todos os dispositivos registrados em uma ID da Amazon específica, é possível usar o filtro de tópico a seguir:

`$aws/iotwireless/events/device_registration_state/+/sidewalk/sidewalk_accounts/amazon_id/+`

> **Note**
>
> Você não pode usar o caractere curinga`#`para se inscrever nos tópicos reservados. Para obter mais informações sobre os filtros de tópicos, consulteFiltros de tópicos (p. 100).

Carga útil da mensagem para eventos de estado de registro de dispositivos

Depois de ativar notificações de eventos de estado de registro de dispositivo, as notificações de eventos são publicadas por meio do MQTT com uma carga JSON. Esses eventos contêm as seguintes cargas de exemplo:

```
{
    "eventId": "string",
    "eventType": "provisioned|registered",
    "WirelessDeviceId": "string",
    "timestamp": "timestamp",

    // Event-specific fields
    "operation": "create|deregister|register",
    "Sidewalk": {
        "AmazonId": "string",
        "SidewalkManufacturingSn": "string"
    }
}
```

A carga contém os seguintes atributos:

eventId

Um ID de evento exclusivo (sequência).

eventType

O tipo de evento que ocorreu. Pode ser `provisioned` ou `registered`.

WirelessDeviceID

O identificador do dispositivo sem fio.

timestamp

A data e hora do Unix de quando o evento ocorreu.

operação

A operação que acionou o evento. Os valores válidos são `create`, `register` e `deregister`.

calçada

O ID da Amazon Sidewalk ou`SidewalkManufacturingSn`Para o qual você deseja receber notificações de eventos.

Eventos de proximidade

Eventos de proximidade publicam notificações de eventosAWS IoTrecebe um farol do dispositivo Sidewalk. Quando o dispositivo Sidewalk se aproxima do Amazon Sidewalk, os beacons enviados do seu dispositivo são filtrados pelo Amazon Sidewalk em intervalos regulares e recebidos porAWS IoT Wireless.AWS IoT Wirelessentão o notifica sobre esses eventos quando um farol é recebido.

Como funcionam os eventos de proximidade

Eventos de proximidade notificam você quandoAWS IoTrecebe um farol, seus dispositivos Sidewalk podem emitir beacons a qualquer momento. Quando seu dispositivo está perto do Amazon Sidewalk, o Sidewalk recebe os beacons e os encaminha paraAWS IoT WirelessEm intervalos de tempo regulares. O Amazon Sidewalk configurou esse intervalo de tempo como 10 minutos. QuandoAWS IoT Wirelessrecebe o farol da Sidewalk, você será notificado do evento.

Eventos de proximidade o notificarão quando um farol for descoberto ou quando um farol for perdido. Você pode configurar os intervalos nos quais você é notificado sobre o evento de proximidade.

Ative notificações para eventos de proximidade

Antes que os assinantes dos tópicos reservados de proximidade do Sidewalk possam receber mensagens, você deve habilitar notificações de eventos para eles a partir doAWS Management Consoleou usando a API ou a CLI do. Você pode ativar esses eventos para todos os recursos do Sidewalk em seuConta da AWSou para recursos selecionados. Para obter informações sobre como ativar esses eventos, consulteAtivar eventos para recursos sem fio (p. 1249).

Formato de tópicos MQTT para eventos de proximidade

Para notificá-lo de eventos de proximidade, você pode se inscrever em tópicos reservados do MQTT que começam com um sinal de dólar ($). Para obter mais informações, consulte Tópicos MQTT (p. 99).

Tópicos MQTT reservados para eventos de proximidade de calçada usam o formato:

- Para tópicos em nível de recurso:

  ```
  $aws/iotwireless/events/{eventName}/{eventType}/sidewalk/wireless_devices
  ```

- Para tópicos de identificador:

  ```
  $aws/iotwireless/events/{eventName}/{eventType}/sidewalk/{resourceType}/
  {resourceID}/{id}
  ```

Onde:

{eventName}

{eventName} deve ser`proximity`.

{eventType}

{eventType} pode ser`beacon_discovered`ou`beacon_lost`.

{resourceType}

{resourceType} pode ser`sidewalk_accounts`ou`wireless_devices`.

{resourceId}

{resourceId} é`amazon_id`para {resourceType} de`sidewalk_accounts`e`wireless_device_id`para {resourceType} de`wireless_devices`.

Você também pode usar o`+`caractere curinga para assinar vários tópicos ao mesmo tempo. O`+`Caractere curinga corresponde a qualquer string no nível que contém o caractere. Por exemplo, caso você queira ser notificado de todos os tipos de evento possíveis (`beacon_discovered`e`beacon_lost`) e para todos os dispositivos registrados em uma ID da Amazon específica, é possível usar o filtro de tópico a seguir:

```
$aws/iotwireless/events/proximity/+/sidewalk/sidewalk_accounts/amazon_id/+
```

Note

Você não pode usar o caractere curinga#para se inscrever nos tópicos reservados. Para obter mais informações sobre os filtros de tópicos, consulteFiltros de tópicos (p. 100).

Carga útil da mensagem para eventos de proximidade

Depois de ativar notificações de eventos de proximidade, as mensagens de eventos são publicadas por meio do MQTT com uma carga JSON. Esses eventos contêm as seguintes cargas de exemplo:

```
{
    "eventId": "string",
    "eventType": "beacon_discovered|beacon_lost",
    "WirelessDeviceId": "string",
    "timestamp": "1234567890123",

    // Event-specific fields
    "Sidewalk": {
        "AmazonId": "string",
        "SidewalkManufacturingSn": "string"
    }
}
```

A carga contém os seguintes atributos:

eventId

Um ID de evento exclusivo, que é uma string.

eventType

O tipo de evento que ocorreu. Pode ser `beacon_discovered` ou `beacon_lost`.

WirelessDeviceID

O identificador do dispositivo sem fio.

timestamp

A data e hora do Unix de quando o evento ocorreu.

calçada

O ID da Amazon Sidewalk ou`SidewalkManufacturingSn`Para o qual você deseja receber notificações de eventos.

Integração do Alexa Voice Service (AVS) aoAWS IoT

Integração do Alexa Voice Service (AVS) aoAWS IoTO é um novo recurso que leva o Alexa Voice a qualquer dispositivo conectado sem incorrercustos de mensagens. AVS paraAWS IoTReduz o custo e a complexidade da integração com a Alexa. Esse recurso utiliza o AWS IoT para descarregar tarefas de áudio de memória e computação intensivas do dispositivo para a nuvem. Devido à redução resultante no custo da lista de materiais de engenharia (eBoM), os fabricantes de dispositivos agora podem trazer a Alexa para dispositivos de IoT com restrições de recursos e possibilitar que os consumidores falem diretamente com a Alexa em cômodos da casa, no escritório ou em quartos de hotel para uma experiência ambiente.

No momento, os dispositivos de IoT de casa inteligente são criados com microcontroladores (MCU) de baixo custo com memória limitada para executar sistemas operacionais em tempo real. Anteriormente, as soluções de AVS para produtos integrados da Alexa exigiam caros dispositivos de aplicativos com base em processador com mais de 50 MB de memória com execução em Linux ou Android. Esses requisitos caros de hardware tornaram dispendiosa a integração do Alexa Voice a dispositivos de IoT restritos a recursos. AVS paraAWS IoTLeva a funcionalidade integrada do Alexa em MCUs, como os processadores série Arm Cortex-M com menos de 1 MB de RAM incorporada. Para fazer isso, o AVS descarrega na nuvem tarefas de memória e computação para um dispositivo virtual com Alexa integrada. Isso reduz o custo da eBOM em até 50%.

Para obter mais informações sobre os processadores série Arm Cortex-M, consulte Arm ou Wikipedia. Para obter mais informações sobre os requisitos de hardware para produtos integrados com a Alexa, consulte Sizing Up CPU, Memory and Storage for Your Alexa Built-in Device no portal do desenvolvedor do Amazon Alexa.

> **Note**
>
> AVS paraAWS IoTO está disponível em todosRegião da AWSs ondeAWS IoTO está disponível, exceto nas regiões da China (Pequim e Ningxia). Para a lista atual deAWSRegiões, veja aAWSTabela de região.

AVS paraAWS IoTO tem três componentes:

- Um conjunto de tópicos de MQTT reservados para transferir mensagens de áudio entre dispositivos compatíveis com a Alexa e o AVS.
- Um dispositivo virtual compatível com a Alexa na nuvem que muda tarefas relacionadas à recuperação de mídia, à decodificação de áudio, à mixagem de áudio e ao gerenciamento de estado do dispositivo físico para o dispositivo virtual.
- Um conjunto de APIs que oferecem suporte ao recebimento e ao envio de mensagens por meio de tópicos reservados, interagindo com o microfone e com o alto-falante do dispositivo e gerenciando o estado do dispositivo.

O diagrama a seguir ilustra como esses componentes funcionam em conjunto. Ele também demonstra como os fabricantes de dispositivos usam o serviço Login with Amazon para autenticar o AVS.

Os fabricantes de dispositivos têm duas opções para começar a usar a Integração do AVS para AWS IoT.

- Kits de desenvolvimento— Kits de desenvolvimento lançados por nossos parceiros facilitam o início do uso. ONXP i.MX RT 106 A,Kit de desenvolvimento Qualcomm Home Hub 100 para Amazon AVS, eSTM32 STEVAL-VOICE-UIsão alguns dos kits disponíveis no mercado. É possível encontrá-los em Kits de desenvolvimento para o AVS. Os kits incluem conectividade pronta para uso com o AWS IoT Core, algoritmos de áudio qualificados para AVS para captação de voz em campo distante, cancelamento de eco, palavra de ativação da Alexa e código de aplicativo do AVS para AWS IoT. É possível usar o código do aplicativo de recursos para criar um protótipo de um dispositivo rapidamente e fazer a portabilidade da implementação para o design de MCU escolhido para testes e produção de dispositivos quando você estiver pronto.
- Código de aplicativo personalizado do lado do dispositivo— Os desenvolvedores também podem escrever um AVS personalizado paraAWS IoTaplicativo usando a API disponível publicamente. A documentação dessa API está disponível na página do desenvolvedor do AVS. Você pode baixar o FreeRTOS eAWS IoTSDK do dispositivo a partir do console FreeRTOS (https://console.aws.amazon.com/freertos/) ouGitHub.

Para ver um exemplo de como começar a usar um kit de desenvolvimento, consulteConceitos básicos do Alexa Voice Service (AVS) Integration aoAWS IoTem um dispositivo NXP.

Conceitos básicos do Alexa Voice Service (AVS) Integration aoAWS IoTem um dispositivo NXP

Com o kit de desenvolvimento NXP i.MX RT106A, você pode visualizar a integração do Alexa Voice Service (AVS) paraAWS IoTUsar uma conta do NXP pré-configurada. Após visualizar a funcionalidade

com a conta do NXP, é possível personalizar o firmware (código-fonte do aplicativo) para usar sua própria conta. Este tópico orienta você durante as etapas para visualizar a conta pré-configurada e para personalizar o dispositivo com sua própria conta.

Tópicos

- Visualizar a integração do Alexa Voice Service (AVS) aoAWS IoTCom uma conta do NXP pré-configurada (p. 1264)
- Interaja com a Alexa (p. 1275)
- Use a suaAWSe contas de desenvolvedor do Alexa Voice Service para configurar o AVS aoAWS IoT (p. 1277)

Visualizar a integração do Alexa Voice Service (AVS) aoAWS IoTCom uma conta do NXP pré-configurada

Pré-requisitos

Para seguir essas etapas, é necessário ter os recursos a seguir.

- Kit de desenvolvimento NXP i.MX RT106A

  Este kit é pré-carregado com software que permite ambosConfiguração zero touch (p. 1275)econfiguração guiada pelo usuário (p. 1265).
- Um computador Mac, Windows 7 ou 10 ou Linux
- Um dispositivo móvel Android ou iOS
- OAplicativo Amazon Alexa para iOSou oAplicativo Amazon Alexa para Android
- Uma conta do Amazon Alexa

Ative o kit de desenvolvimento

Verifique se a caixa do kit de desenvolvimento contém um cabo USB tipo C para duplo tipo A. Conecte ambas as conexões USB-A ao computador. Conecte o conector USB-C ao kit. Sua configuração será exibida como a imagem a seguir.

> **Note**
>
> Se a caixa contiver um cartão de início rápido, desconsidere e consulte essas instruções.

Quando a placa tem energia, o LED indicador de status acende e exibe várias cores. Esses são indicadores de status para os vários estágios do processo de inicialização. As cores e a taxa de piscar indicam o status do dispositivo. Seu dispositivo está pronto para configuração quando a luz indicadora de status ficar azul sólido, conforme mostrado na imagem a seguir.

O kit de desenvolvimento suporta as seguintes configurações, dependendo do seu ambiente.

- Configuração guiada pelo usuário (p. 1265): Use essa configuração quando o dispositivo chegar no estado de fábrica e não atender às condições para a configuração de zero touch (ZTS).

  Você também usa a configuração guiada pelo usuário quando alguém já executou o ZTS no dispositivo. O ZTS pode ocorrer apenas uma vez na vida útil de um produto.
- Configuração de toque zero (ZTS) (p. 1275): Use essa configuração quando o ambiente atender às condições a seguir.
  - Você comprou o kit da Amazon.com.
  - Você não comprou o kit nem recebeu o kit como presente.
  - Você já instalou um dispositivo de provisionador na rede Wi-Fi que está usando o kit.

    Um dispositivo de provisionador é um dispositivo da Amazon (como um Echo (3ª geração)) registrado em uma conta de cliente da Amazon.

    Para obter uma lista de dispositivos da Amazon que se qualificam como dispositivos de provisionamento, consulteTestando seu dispositivoemCompreendendo a configuração sem frustração.
  - Seu kit está dentro da faixa Bluetooth Low Energy (BLE) do dispositivo provisionador.
  - Suas credenciais de Wi-Fi estão disponíveis no armário Amazon Wi-Fi.
  - Você tem umHabilidade do Alexavinculado à sua conta da Amazon.
  - Você implementouLogin da Amazon.

  Para obter mais informações sobre esse tipo de configuração, consulteConfiguração do Zero Touch.

Configuração guiada pelo usuário

Quando um kit que não atende aos requisitos do ZTS é ativado, ele aguarda que a configuração guiada pelo usuário ocorra por meio do aplicativo Amazon Alexa em seu telefone. Certifique-se de que o aplicativo

Amazon Alexa esteja instalado em seu telefone e se as permissões de Bluetooth e localização estão ativadas para o aplicativo.

O procedimento a seguir descreve como executar a configuração guiada pelo usuário.

1. Abra o aplicativo Alexa e faça login na sua conta do Amazon Alexa. O aplicativo detecta que um dispositivo próximo está aguardando a configuração guiada pelo usuário e exibe a página na imagem a seguir. Escolha Continue.

NXP IoT development kit can be set up
LATER
CONTINUE
to try.
Home
Communicate
Play
Devices
More

Se escolherMais tardeou se o aplicativo não exibir esta página, use as etapas a seguir para iniciar a configuração guiada pelo usuário.

1. Selecione oDispositivose, depois, selecione o sinal de adição (+) na janela que aparece.
2. SelecioneAdicionar dispositivo.
3. SelecioneDispositivo de desenvolvimento.
4. NoQual marca é o seu dispositivo de desenvolvimento?Página, escolhaNXPe, depois, escolhaPróximo.

   As imagens a seguir mostram como os prompts descritos nessas etapas aparecem no aplicativo.

Quando o aplicativo se conecta ao dispositivo, a luz indicadora de status pisca em laranja, como nas imagens a seguir.

Note

Se a configuração guiada pelo usuário for interrompida (por exemplo, se você fechar o aplicativo), o dispositivo retornará ao modo de descoberta e a luz indicadora de status exibirá azul sólido.

2. O aplicativo pede que o kit escaneie o ambiente em busca de redes Wi-Fi e retorne uma lista de redes que ele detecta. Escolha a rede à qual o dispositivo deve se conectar. A imagem a seguir mostra como essa lista aparece no aplicativo.

> **Note**
>
> Se você já salvou a rede selecionada selecionada em sua conta da Amazon, não será necessário digitar a senha do Wi-Fi.

Quando você seleciona a rede Wi-Fi, a tela exibe a seguinte mensagem à medida que o provisionamento Wi-Fi e a comunicação com os servidores de configuração ocorrem: Conectando seu dispositivo de desenvolvimento NXP ao***Nome da rede Wi-Fi***. A imagem a seguir mostra como essa tela aparece no aplicativo.

A luz indicadora de status continua piscando em laranja até que o registro do kit esteja concluído. Quando o registro estiver concluído, o dispositivo diz: “Seu dispositivo Alexa está pronto”. O kit então reinicia.

A seguir descreve as etapas que o kit executa após reiniciar e se reconectar à rede Wi-Fi selecionada.

1. À medida que ele reinicia, o kit exibe novamente várias cores e alterna entre piscas e cores sólidas à medida que avança no processo de inicialização.
2. O dispositivo então tenta se reconectar à rede Wi-Fi selecionada. Ao fazer isso, a luz indicadora de status pisca amarela em intervalos de 500 milissegundos (ms). Depois de se conectar à rede Wi-Fi, ele pisca amarelo mais rápido, em intervalos de 250 ms. As imagens a seguir mostram como esse piscar aparece no kit.

3. O kit se conecta aAWS IoT. Enquanto ele se conecta, a luz indicadora de status pisca em verde em intervalos de 500 ms. Quando o kit terminar de conectar, a luz indicadora de status pisca em verde em intervalos de 250 ms. As imagens a seguir mostram como esse piscar aparece no kit.

4. O kit reproduz um som de carrilhão que indica que você pode usá-lo para interagir com a Alexa.

Quando o kit se conecta aAWS IoT, a tela na imagem a seguir aparece no aplicativo.

SETUP
NXP light connected
NXP Development Kit has been added to your Alexa account. Next, let's continue setup.
NEXT

OLuz NXP conectadamensagem aparece no aplicativo porque o kit implementa os recursos de casa inteligente para um dispositivo de luz NXP.

Configuração de toque zero (ZTS)

Se o seu ambiente atender a todos os pré-requisitos do ZTS, o dispositivo de provisionamento detectará seu kit e iniciará a configuração do ZTS quando você liga o kit. O aplicativo Amazon Alexa também é um provisionador ZTS, portanto, abrir o aplicativo Amazon Alexa também pode iniciar a configuração do ZTS.

À medida que o processo de provisionamento continua, o estado da luz indicadora de status segue os mesmos padrões descritos na seção de configuração guiada pelo usuário. Durante o provisionamento, as mensagens de log são enviadas para o console SLN-ALEXA-IOT por meio de sua porta COM virtual. Quando o provisionamento estiver concluído, o kit reproduz o som do chime que indica que você pode usá-lo para interagir com a Alexa.

> **Note**
>
> A configuração do ZTS só pode ocorrer uma vez na vida útil de um dispositivo, mesmo que você o retorne às configurações de fábrica.

Interaja com a Alexa

Você pode começar a usar o kit para interagir com a Alexa fazendo uma pergunta. Mesmo uma pergunta simples, como “Alexa, como está o clima?” passa por vários estados à medida que a Alexa processa e responde a ele.

Você vê a primeira indicação de que o kit está ouvindo quando você fala a palavra de ativação da Alexa. Quando o kit detecta essa palavra, o kit começa a ouvir e enviar informações do microfone para o AVS atravésAWS IoT. A luz indicadora de status exibe ciano sólido, como na imagem a seguir.

Quando o dispositivo terminar de enviar informações do microfone para o AVS atravésAWS IoT, o dispositivo pára de ouvir e muda para um estado de pensamento. Esse estado indica que o AVS está processando a pergunta e está determinando a melhor resposta. Enquanto o kit estiver nesse estado, o LED indicador de status pisca ciano e azul em intervalos de 200 ms. As imagens a seguir mostram como esse piscar aparece no kit.

Quando o dispositivo termina de pensar, ele começa a responder. Antes que o kit comece a falar, a luz indicadora de status muda para um estado de fala. O kit pisca ciano e azul em intervalos de 500 ms.

A resposta da Alexa é exibida fora do alto-falante do kit enquanto a luz indicadora de status pisca ciano e azul, a Alexa descreve as condições meteorológicas com base na localização da sua conta de consumidor Alexa. Quando a resposta estiver concluída, a luz indicadora de status pára de piscar e desliga. Isso indica que o kit está em estado ocioso e aguardando a palavra de ativação Alexa.

Use a suaAWSe contas de desenvolvedor do Alexa Voice Service para configurar o AVS aoAWS IoT

A conta NXP pré-configurada é apenas para avaliar o kit. Ao usar sua própria conta, você tem os seguintes benefícios.

- Controle total de trabalhos e implantações sobre o ar (OTA), como atualizações remotas de firmware.
- Controle sobreAWSServiços da .
- Personalização de habilidades em casa inteligente.

Para migrar da conta NXP pré-configurada para sua própria conta, baixe oGuia de migração da solução de voz MCU AlexadoConceitos básicosSeção do doSolução baseada em MCU EdgeReady para Alexa para IOT. Siga as etapas deste guia.

> **Note**
>
> Para baixar esse arquivo, você precisa de uma conta NXP.

AWS IoTSDKs de dispositivos, SDKs móveis eAWS IoTCliente de dispositivo

Esta página resume oAWS IoTSDKs de dispositivos, bibliotecas de código aberto, guias do desenvolvedor, aplicativos de amostra e guias de portabilidade para ajudá-lo a criar soluções inovadoras de IoT comAWS IoTe sua escolha de plataformas de hardware.

Esses SDKs são para uso em seu dispositivo IoT. Se você estiver desenvolvendo um aplicativo de IoT para uso em um dispositivo móvel, consulte oSDKs móveis do AWS (p. 1280). Se você estiver desenvolvendo um aplicativo IoT ou programa do lado do servidor, consulte oAWS SDKs (p. 74).

SDKs de dispositivo da AWS IoT

Os SDKs de dispositivos da AWS IoT incluem bibliotecas de código aberto, guias de desenvolvedor com exemplos e guias de portabilidade para que você possa criar produtos ou soluções inovadoras da IoT nas plataformas de hardware de sua preferência.

Esses SDKs ajudam você a conectar seus dispositivos IoT aAWS IoTusando os protocolos MQTT e WSS.

C++

AWS IoTSDK de dispositivo C++ da

OAWS IoTO SDK de dispositivo C++ permite que os desenvolvedores compilem aplicativos conectados usandoAWSO e aAWS IoTAPIs. Esse SDK foi especificamente projetado para dispositivos que não têm restrições de recursos e exigem recursos avançados, como enfileiramento de mensagens, suporte a vários threads e os mais recentes recursos de linguagem. Para mais informações, consulte:

- AWS IoTSDK de dispositivo C++ v2 no GitHub
- AWS IoTLeia-me do SDK de dispositivo C++ v2
- AWS IoTAmostras do SDK de dispositivo C++ v2
- AWS IoTDocumentação da API do SDK C++ v2 do dispositivo

Python

AWS IoTSDK de dispositivo da para Python

OAWS IoTO SDK de dispositivo para Python permite que os desenvolvedores escrevam scripts Python para usar seus dispositivos para acessar oAWS IoTplataforma através de MQTT ou MQTT sobre o WebSocket protocolo. Conectando seus dispositivos aAWS IoT, os usuários podem trabalhar de modo seguro com o agente de mensagens, as regras e o Shadow fornecidos pelaAWS IoTe com outrosAWSserviços comoAWS Lambda, Kinesis e Amazon S3 e muito mais.

- SDK de dispositivo AWS IoT para Python v2 no GitHub

- AWS IoT Device SDK for Python v2 Readme
- AWS IoTSDK de dispositivo para amostras do Python v2
- AWS IoTDocumentação da API Device SDK for Python v2

JavaScript

AWS IoTSDK de dispositivo da para JavaScript

O pacote aws-iot-device-sdk.js permite que os desenvolvedores escrevam JavaScript aplicativos que acessamAWS IoTusando MQTT ou MQTT sobre o WebSocket protocolo. Ele pode ser usado em ambientes Node.js e aplicações de navegador. Para obter mais informações, consulte as informações a seguir:

- AWS IoTSDK de dispositivo da para JavaScript v2 no GitHub
- AWS IoTSDK de dispositivo da para JavaScript Leia-me do v2
- AWS IoTSDK de dispositivo da para JavaScript Amostras do v2
- AWS IoTSDK de dispositivo da para JavaScript Documentação da API v2

Java

AWS IoTSDK de dispositivo da para Java

OAWS IoTO SDK de dispositivo para Java permite que os desenvolvedores de Java acessem oAWS IoTplataforma através de MQTT ou MQTT sobre o WebSocket protocolo. O SDK é criado com suporte a sombras. Você pode acessar as sombras usando métodos HTTP, inclusive GET, UPDATE e DELETE. O SDK também oferece suporte a um modelo simplificado de acesso a sombras, o que permite que os desenvolvedores troquem dados com as sombras usando apenas os métodos getter e setter, sem necessidade de serializar ou desserializar nenhum documento JSON. Para obter mais informações, consulte as informações a seguir:

- SDK de dispositivo AWS IoT para Java v2 no GitHub
- Leia-me do SDK de dispositivo AWS IoT para Java v2
- AWS IoTExemplos do SDK de dispositivo para Java v2
- AWS IoTDocumentação da API do SDK de dispositivo para Java v2

SDK de dispositivo da AWS IoT para C incorporado

Note

Este SDK destina-se ao uso por desenvolvedores experientes de software incorporado.

OAWS IoT Device SDK para C incorporado(C-SDK) é um conjunto de arquivos de origem C sob a licença de código aberto do MIT que podem ser usados em aplicativos incorporados para conectar dispositivos de IoT com segurançaAWS IoT Core. Ele inclui um cliente MQTT, JSON Parser eAWS IoTDevice Shadow da,AWS IoTTrabalhos,AWS IoTProvisionamento de frota eAWS IoT Device DefenderBibliotecas do . Esse SDK é distribuído na forma de origem e pode ser incorporado ao firmware do cliente junto com o código do aplicativo, outras bibliotecas e um sistema operacional (SO) de sua escolha.

Em geral, o AWS IoT Device SDK para C incorporado destina-se a dispositivos com restrição de recursos que exigem um tempo de execução de linguagem C otimizado. É possível usar o SDK em qualquer sistema operacional e hospedá-lo em qualquer tipo de processador (p. ex., MCUs e MPUs).

Para obter mais informações, consulte as informações a seguir:

- SDK de dispositivo AWS IoT para C incorporado no GitHub

- Arquivo leiame do SDK de dispositivo da AWS IoT para C incorporado
- AWS IoTSDK de dispositivo da para amostras C incorporadas

AnteriormenteAWS IoTVersões dos SDKs do

Essas são as versões anteriores doAWS IoTSDKs de dispositivo que foram substituídos pelas versões mais recentes listadas acima. Esses SDKs estão recebendo apenas atualizações de manutenção e segurança. Eles não serão atualizados para incluir novos recursos e não devem ser usados em novos projetos.

- SDK de dispositivo C++ AWS IoT no GitHub
- Arquivo leiame do SDK de dispositivo C++ da AWS IoT
- AWS IoT Device SDK for Python v1 on GitHub
- AWS IoT Device SDK for Python v1 Readme
- SDK de dispositivo da AWS IoT para Java no GitHub
- Leiame do SDK para Java do dispositivo do AWS IoT
- AWS IoTSDK de dispositivo da para JavaScript no GitHub
- AWS IoTSDK de dispositivo da para JavaScript Readme (Leiame)
- Arduino Yún SDK no GitHub
- Arquivo Readme do Arduino Yún SDK

SDKs móveis do AWS

OAWSOs SDKs móveis oferecem suporte específico da plataforma para desenvolvedores de aplicativos móveis para as APIs doAWS IoT Coreserviços, comunicação de dispositivos IoT usando MQTT e as APIs de outrosAWSServiços da .

Android

AWS Mobile SDK for Android

OAWS Mobile SDK for Androidcontém uma biblioteca, exemplos e documentação para que os desenvolvedores compilem aplicativos móveis conectados usandoAWS. Esse SDK também inclui suporte para comunicações de dispositivo MQTT e chamar as APIs doAWS IoT CoreServiços da . Para obter mais informações, consulte as informações a seguir:

- AWS Mobile SDK for Android no GitHub
- AWS Mobile SDK for AndroidReadme (Leiame)
- AWS Mobile SDK for Android Samples
- Referência da API do AWS Mobile SDK for Android
- Documentação de referência da AwSiotClient

iOS

AWS Mobile SDK for iOS

OAWS Mobile SDK for iOSÉ um kit de desenvolvimento de software de código aberto distribuído sob uma licença Apache Open Source. OAWS Mobile SDK for iOSFornece uma biblioteca, exemplos de código e documentação para ajudar os desenvolvedores a criar aplicativos móveis conectados usandoAWS. Esse SDK também inclui suporte para comunicações de dispositivo MQTT e chamar as APIs doAWS IoT CoreServiços da . Para obter mais informações, consulte as informações a seguir:

- AWS Mobile SDK for iOS no GitHub
- AWS Mobile SDK for iOSReadme (Leiame)
- AWS Mobile SDK for iOS Samples
- Documentos de referência da classe AWSiot noAWS Mobile SDK for iOS

AWS IoTCliente de dispositivo

OAWS IoTO Device Client fornece código para ajudar o dispositivo a se conectar aoAWS IoT, execute tarefas de provisionamento de frota, ofereça suporte a políticas de segurança de dispositivos, conecte-se usando tunelamento seguro e processe trabalhos em seu dispositivo. Você pode instalar esse software em seu dispositivo para lidar com essas tarefas de rotina do dispositivo para que você possa se concentrar em sua solução específica.

> **Note**
>
> OAWS IoTO Device Client funciona com dispositivos IoT baseados em microprocessador com processadores x86_64 ou ARM e sistemas operacionais Linux comuns.

C++

AWS IoTCliente de dispositivo

Para obter mais informações sobre oAWS IoTDevice Client em C++, consulte o seguinte:

- AWS IoTCliente de dispositivo no código-fonte C++ no GitHub
- AWS IoTCliente de dispositivo em C++ Readme

Resolução de problemas AWS IoT

Ajude-nos a melhorar esse tópico

Deixe-nos saber o que ajudaria a torná-lo melhor

As informações a seguir podem ajudar a solucionar problemas comuns no AWS IoT.

Tarefas

- Diagnóstico de problemas de conectividade (p. 1282)
- Diagnosticar problemas de regras (p. 1285)
- Diagnosticar problemas com shadows (p. 1286)
- Diagnosticar problemas de ação no fluxo de entrada da Salesforce IoT (p. 1287)
- Guia de solução de problemas de indexação de frota (p. 1288)
- Solução de problemas “Limite de fluxo excedido para oAWSconta” (p. 1290)
- Guia de solução de problemas do AWS IoT Device Defender (p. 1290)
- AWS IoTGuia de solução de problemas do Device Advisor (p. 1294)
- Solução de problemas de desconexões de frota de dispositivos (p. 1295)
- Erros do AWS IoT (p. 1296)

Diagnóstico de problemas de conectividade

Ajude-nos a melhorar esse tópico

Deixe-nos saber o que ajudaria a torná-lo melhor

Uma conexão bem-sucedida comAWS IoTO requer:

- Uma conexão válida
- Um certificado válido e ativo
- Uma política que permite a conexão e a operação desejadas

Conexão

Como encontro o endpoint correto?

- O `endpointAddress` retornado por `aws iot describe-endpoint --endpoint-type iot:Data-ATS`

  ou
- O `domainName` retornado por `aws iot describe-domain-configuration --domain-configuration-name "domain_configuration_name"`

Como faço para encontrar o valor correto da Server Name Indication (SNI)?

O valor SNI correto é `oendpointAddress`retornado pelodescribe-endpointoudescribe-domain-configurationComandos do . É o mesmo endereço do endpoint na etapa anterior.

Como soluciono um problema de conectividade que persiste?

Você pode usar oAWSDevice Advisor para ajudar a solucionar problemas. Os testes pré-criados do Device Advisor ajudam você a validar o software do dispositivo em relação às melhores práticas para o uso deTLS,MQTT,AWS IoTDevice Shadow, eAWS IoTTrabalhos do.

Aqui está um link para o existenteDevice Advisorconteúdo.

Autenticação

Os dispositivos devem serautenticada (p. 297)Para conectar o aoAWS IoTEndpoints do . Para dispositivos que usamCertificados do cliente X.509 (p. 298)para autenticação, os certificados devem ser registrados comAWS IoTe seja ativo.

Como meus dispositivos autenticam endpoints da AWS IoT?

Adicione o certificado CA da AWS IoT ao armazenamento de confiança do seu cliente. Consulte a documentação do noAutenticação do servidor noAWS IoT CoreE siga os links para fazer download do certificado apropriado da CA.

O que é verificado quando um dispositivo se conecta aoAWS IoT?

Quando um dispositivo tenta se conectar aoAWS IoT:

1. AWS IoTVerifique se há um certificado válido e valor SNI (Server Name Indication, indicação de nome de servidor).
2. AWS IoTverifica se o certificado usado está registrado com oAWS IoTConta e que ela foi ativada.
3. Quando um dispositivo tenta executar qualquer ação noAWS IoT, como assinar ou publicar uma mensagem, a política anexada ao certificado usado para se conectar é marcada para confirmar se o dispositivo está autorizado a executar essa ação.

Como posso validar um certificado configurado corretamente?

Use o comando `s_client` OpenSSL para testar uma conexão com o endpoint da AWS IoT:

```
openssl s_client -connect custom_endpoint.iot.aws-region.amazonaws.com:8443 -
CAfile CA.pem -cert cert.pem -key privateKey.pem
```

Para obter mais informações sobre como usar o `openssl s_client`, consulte a documentação do OpenSSL s_client.

Como faço para verificar o status de um certificado?

- Listar os certificados

  Se você não souber o ID do certificado, poderá ver o status de todos os seus certificados usando `aws iot list-certificates`comando.

- Mostrar os detalhes de um certificado

  Se você souber o ID do certificado, esse comando mostrará informações mais detalhadas sobre o certificado.

  ```
  aws iot describe-certificate --certificate-id "certificateId"
  ```

- Revise o certificado noAWS IoTConsole do

  NoAWS IoTconsoleNo menu esquerdo, escolhaSecuree, depois, escolhaCertificados.

  Escolha o certificado que você está usando para se conectar na lista para abrir sua página de detalhes.

Na página de detalhes do certificado, você pode ver o status atual.

O status do certificado pode ser alterado usando oAçõesno canto superior direito da página de detalhes.

Autorização

AWS IoTUso de recursos doPolíticas do AWS IoT Core (p. 332)para autorizar esses recursos a executarações (p. 333). Para que uma ação seja autorizada, o especificadoAWS IoTos recursos devem ter um documento de política anexado a ele que conceda permissão para executar essa ação.

Recebi uma resposta `PUBNACK` ou `SUBNACK` do operador. O que devo fazer?

Verifique se há uma política anexada ao certificado que você está usando para chamar a AWS IoT. Todas as operações de publicação/assinatura são negadas por padrão.

Verifique se a política anexada autoriza oações (p. 333)Você está tentando executar o.

Verifique se a política anexada autoriza orecursos (p. 335)que estão tentando executar as ações autorizadas.

Eu tenho umAUTHORIZATION_FAILUREentrada em meus registros.

Verifique se há uma política anexada ao certificado que você está usando para chamar a AWS IoT. Todas as operações de publicação/assinatura são negadas por padrão.

Verifique se a política anexada autoriza oações (p. 333)Você está tentando executar o.

Verifique se a política anexada autoriza orecursos (p. 335)que estão tentando executar as ações autorizadas.

Como verifico o que a política autoriza?

NoAWS IoTconsoleNo menu esquerdo, escolhaSecuree, depois, escolhaCertificados.

Escolha o certificado que você está usando para se conectar na lista para abrir sua página de detalhes.

Na página de detalhes do certificado, você pode ver o status atual.

No menu à esquerda da página de detalhes do certificado, escolhaPolíticasPara ver as políticas anexadas ao certificado.

Escolha a política desejada para ver a página de detalhes.

Na página de detalhes da política, revise a políticaDocumento de políticapara ver o que ele autoriza.

SelecioneEditar documento de políticaPara fazer as alterações no documento de política.

Segurança e identidade

Quando você fornece os certificados do servidor paraAWS IoTconfiguração de domínio personalizado, os certificados têm no máximo quatro nomes de domínio.

Para obter mais informações, consulte Endpoints e cotas do AWS IoT Core.

Diagnosticar problemas de regras

Ajude-nos a melhorar esse tópico

Deixe-nos saber o que ajudaria a torná-lo melhor

Esta seção descreve algumas das coisas a serem verificadas quando você encontra um problema com a regra.

Configurar o CloudWatch Logs para solução de problemas

O uso do é a melhor maneira de depurar os problemas que você está tendo com regras é usar CloudWatch Logs. Quando você habilita CloudWatch Logs do para oAWS IoT, você pode ver quais regras são acionadas, bem como o sucesso ou a falha delas. Você também obtém informações que indicam se as condições da cláusula WHERE são correspondentes. Para obter mais informações, consulte MonitorAWS IoTusando CloudWatch Logs (p. 454).

O problema mais comum de regras é a autorização. Os logs mostram se sua função não está autorizada a executar AssumeRole no recurso. Este é um log de exemplo gerado por log refinado (p. 437):

```
{
    "timestamp": "2017-12-09 22:49:17.954",
    "logLevel": "ERROR",
    "traceId": "ff563525-6469-506a-e141-78d40375fc4e",
    "accountId": "123456789012",
    "status": "Failure",
    "eventType": "RuleExecution",
    "clientId": "iotconsole-123456789012-3",
    "topicName": "test-topic",
    "ruleName": "rule1",
    "ruleAction": "DynamoAction",
    "resources": {
        "ItemHashKeyField": "id",
        "Table": "trashbin",
        "Operation": "Insert",
        "ItemHashKeyValue": "id",
        "IsPayloadJSON": "true"
    },
    "principalId": "ABCDEFG1234567ABCD890:outis",
    "details": "User: arn:aws:sts::123456789012:assumed-role/dynamo-testbin/5aUMInJH
 is not authorized to perform: dynamodb:PutItem on resource: arn:aws:dynamodb:us-
east-1:123456789012:table/testbin (Service: AmazonDynamoDBv2; Status Code: 400; Error Code:
 AccessDeniedException; Request ID: AKQJ987654321AKQJ123456789AKQJ987654321AKQJ987654321)"
}
```

Este é um log de exemplo semelhante gerado por log global (p. 435):

```
2017-12-09 22:49:17.954 TRACEID:ff562535-6964-506a-e141-78d40375fc4e
PRINCIPALID:ABCDEFG1234567ABCD890:outis [ERROR] EVENT:DynamoActionFailure
TOPICNAME:test-topic CLIENTID:iotconsole-123456789012-3
MESSAGE:Dynamo Insert record failed. The error received was User:
 arn:aws:sts::123456789012:assumed-role/dynamo-testbin/5aUMInJI is not authorized to
 perform: dynamodb:PutItem on resource: arn:aws:dynamodb:us-east-1:123456789012:table/
testbin
(Service: AmazonDynamoDBv2; Status Code: 400; Error Code: AccessDeniedException; Request
 ID: AKQJ987654321AKQJ987654321AKQJ987654321AKQJ987654321).
```

```
Message arrived on: test-topic, Action: dynamo, Table: trashbin, HashKeyField: id,
 HashKeyValue: id, RangeKeyField: None, RangeKeyValue: 123456789012
No newer events found at the moment. Retry.
```

Para obter mais informações, consulte the section called “Visualizar oAWS IoTlogs no CloudWatch console” (p. 454).

Diagnóstico de serviços externos

Os serviços externos são controlados pelo usuário final. Antes da execução da regra, verifique se os serviços externos que você vinculou à sua regra estão configurados e tenham unidades de taxa de transferência suficiente e capacidade para o aplicativo.

Diagnóstico de problemas de SQL

Se a consulta SQL não estiver retornando os dados esperados:

- Revise os registros em busca de mensagens de erro.
- Confirme se a sintaxe SQL corresponde ao documento JSON na mensagem.

  Revise os nomes de objetos e propriedades usados na consulta com aqueles usados no documento JSON da carga útil da mensagem do tópico. Para obter mais informações sobre a formatação JSON em consultas SQL, consulteExtensões JSON (p. 626).
- Verifique se os nomes de objetos ou propriedades JSON incluem caracteres reservados ou numéricos.

  Para obter mais informações sobre caracteres reservados em referências de objetos JSON em consultas SQL, consulteExtensões JSON (p. 626).

Diagnosticar problemas com shadows

Ajude-nos a melhorar esse tópico

Deixe-nos saber o que ajudaria a torná-lo melhor

Diagnosticar shadows

Problema	Diretrizes de solução de problemas
O documento de shadow de um dispositivo é rejeitado com `Invalid JSON document`.	Se você não está familiarizado com o JSON, modifique os exemplos fornecidos neste guia para o seu próprio uso. Para obter mais informações, consulte Exemplos de documentos de sombra (p. 672).
Enviei o JSON correto, mas apenas partes dele (ou nenhuma parte) estão armazenadas no documento de shadow de dispositivo.	Verifique se você está seguindo as diretrizes de formatação JSON. Somente os campos JSON nas seções `desired` e `reported` são armazenados. Conteúdo JSON (mesmo que formalmente correto) fora dessas seções é ignorado.
Eu recebi uma mensagem de erro informando que a shadow de dispositivo excede o tamanho permitido.	O Device Shadow oferece suporte somente a 8 KB de dados. Tente reduzir os nomes de campos dentro do seu documento JSON ou simplesmente criar mais shadows criando mais coisas. Um

Problema	Diretrizes de solução de problemas
	dispositivo pode ter um número ilimitado de coisas/shadows associadas a ele. O único requisito é que cada nome de coisa seja exclusivo na sua conta.
Quando eu recebo uma shadow de dispositivo, ela é maior que 8 KB. Como isso pode acontecer?	Após o recebimento, o serviço da AWS IoT adiciona metadados à shadow do dispositivo. O serviço inclui esses dados em sua resposta, mas não é contabilizado para o limite de 8 KB. Somente os dados dos estados `desired` e `reported` dentro do documento de estado enviado à shadow do dispositivo são contabilizados para o limite.
Minha solicitação foi rejeitada devido à versão incorreta. O que devo fazer?	Execute uma operação GET para sincronizar com a versão mais recente do documento de estado. Ao usar MQTT, assine o tópico /update/accepted para ser notificado sobre alterações no estado e receber a versão mais recente do documento JSON.
O carimbo de data e hora é desativado por alguns segundos.	O carimbo de data e hora para campos individuais e todo o documento JSON são atualizados quando o documento é recebido pelo serviço da AWS IoT ou quando o documento de estado é publicado na mensagem ./update/accepted e ./update/delta. As mensagens podem ser atrasadas na rede, o que pode fazer com que o carimbo de data e hora fique desativado por alguns segundos.
Meu dispositivo pode publicar e se inscrever nos tópicos correspondentes do Shadow, mas quando eu tento atualizar o documento do Shadow pela API REST HTTP, recebo o erro HTTP 403.	Crie políticas no IAM para permitir o acesso a esses tópicos e para a ação correspondente (UPDATE/GET/DELETE) para as credenciais que você está usando. As políticas do IAM e as políticas de certificado são independentes.
Outros problemas.	O serviço Device Shadow registra erros no CloudWatch Logs. Para identificar problemas de configuração e dispositivo, ative o CloudWatch Registra e visualize os logs para obter informações de depuração.

Diagnosticar problemas de ação no fluxo de entrada da Salesforce IoT

Ajude-nos a melhorar esse tópico

Deixe-nos saber o que ajudaria a torná-lo melhor

Rastreamento da execução

Como posso ver o rastreamento da execução de uma ação da Salesforce?

Consulte a seção MonitorAWS IoTusando CloudWatch Logs (p. 454). Depois que os logs forem ativados, você poderá ver o rastreamento da execução da ação da Salesforce.

Sucesso e falha da ação

Como faço para verificar se as mensagens foram enviadas com sucesso para um fluxo de entrada da Salesforce IoT?

Veja os logs gerados pela execução da ação da Salesforce em CloudWatch Logs. Se você vir `Action executed successfully`, isso significará que o mecanismo de regras do AWS IoT recebeu a confirmação da Salesforce IoT de que a mensagem foi enviada com êxito para o fluxo de entrada de destino.

Se houver problemas com a plataforma da Salesforce IoT, entre em contato com o suporte da Salesforce IoT.

O que eu faço se as mensagens não foram enviadas com sucesso para o fluxo de entrada da Salesforce IoT?

Veja os logs gerados pela execução da ação da Salesforce em CloudWatch Logs. Dependendo da entrada do log, você pode tentar as seguintes ações:

`Failed to locate the host`

Verifique se o parâmetro `url` da ação está correto e se o fluxo de entrada da Salesforce IoT existe.

`Received Internal Server Error from Salesforce`

Tentar novamente. Se o problema persistir, entre em contato com o suporte da Salesforce IoT.

`Received Bad Request Exception from Salesforce`

Verifique se há erros na carga que você está enviando.

`Received Unsupported Media Type Exception from Salesforce`

A Salesforce IoT não oferece suporte a uma carga binária no momento. Verifique se você está enviando uma carga JSON.

`Received Unauthorized Exception from Salesforce`

Verifique se o parâmetro `token` da ação está correto e se o token ainda é válido.

`Received Not Found Exception from Salesforce`

Verifique se o parâmetro `url` da ação está correto e se o fluxo de entrada da Salesforce IoT existe.

Se você receber um erro não listado aqui, entre em contato com oAWS IoTSupport ao.

Guia de solução de problemas de indexação de frota

Ajude-nos a melhorar esse tópico

Deixe-nos saber o que ajudaria a torná-lo melhor

Solução de problemas de consultas de agregação para o serviço de indexação de frota

Se você estiver tendo erros de incompatibilidade de tipo, você pode usar CloudWatch Logs para solucionar o problema. CloudWatch Os logs devem ser habilitados antes que os logs sejam gravados pelo serviço

de indexação de frota. Para obter mais informações, consulte MonitorAWS IoTusando CloudWatch Logs (p. 454).

Quando faz consultas de agregação em campos não gerenciados, você só pode especificar um campo definido no argumento `customFields` passado para `UpdateIndexingConfiguration` ou update-indexing-configuration. Se o valor do campo for inconsistente com o tipo de dados de campo configurado, esse valor será ignorado quando você executar uma consulta de agregação.

O serviço de indexação de frota emite um log de erros para CloudWatch Registra quando um campo não pode ser indexado devido a um tipo incompatível. O log de erros contém o nome do campo, o valor que não foi convertido e o nome da coisa para o dispositivo. Veja a seguir um exemplo de log de erros:

```
{
  "timestamp": "2017-02-20 20:31:22.932",
  "logLevel": "ERROR",
  "traceId": "79738924-1025-3a00-a669-7bec69f7f07a",
  "accountId": "000000000000",
  "status": "SucceededWithIssues",
  "eventType": "IndexingCustomFieldFailed",
  "thingName": "thing0",
  "failedCustomFields": [
    {
      "Name": "attributeName1",
      "Value": "apple",
      "ExpectedType": "String"
    },
    {
      "Name": "attributeName2",
      "Value": "2",
      "ExpectedType": "Boolean"
    }
  ]
}
```

Se um dispositivo foi desconectado por aproximadamente uma hora, o valor `timestamp` do status de conectividade pode estar ausente. Para sessões persistentes, o valor pode estar ausente depois que um cliente tiver permanecido desconectado por mais tempo do que o configurado time-to-live (TTL) para a sessão persistente. Os dados de status de conectividade são indexados apenas para conexões em que o ID do cliente tem um nome de coisa correspondente. (O ID do cliente é o valor usado para conectar um dispositivo ao AWS IoT Core.)

Solução de problemas de métricas de frota

Não é possível criar uma métrica de frota

Não há suporte para fazer downgrade de fontes de dados atualizando a configuração de indexação da frota.

Se você tentar criar uma métrica de frota com fontes de dados rebaixadas (por exemplo, anteriormente, as fontes de dados eram dados de registro, dados de sombra e dados de conectividade do dispositivo, e agora as fontes de dados são dados de registro e dados de sombra e sem dados de conectividade do dispositivo), você verá erros e não será possível Crie uma métrica de frota.

A modificação de campos personalizados usados pelas métricas de frota existentes não é suportada.

Não é possível ver pontos de dados no CloudWatch

Se você conseguir criar uma métrica de frota, mas não conseguir ver pontos de dados no CloudWatch, é provável que você não tenha uma coisa que atenda aos critérios de string de consulta.

Veja este exemplo de comando de como criar uma métrica de frota:

```
aws iot create-fleet-metric --metric-name "example_FM" --query-string
 "thingName:TempSensor* AND attributes.temperature>80" --period 60 --aggregation-field
 "attributes.temperature" --aggregation-type name=Statistics,values=count
```

Se você não tiver uma coisa que atenda aos critérios de string de consulta`--query-string "thingName:TempSensor* AND attributes.temperature>80"`:

- com`values=count`, você poderá criar uma métrica de frota e haverá pontos de dados a serem exibidos no CloudWatch. Os pontos de dados do valor`count`é sempre 0.
- com`values`diferentes de`count`, você poderá criar uma métrica de frota, mas não verá a métrica da frota em CloudWatch e não haverá pontos de dados a serem exibidos no CloudWatch.

Solução de problemas “Limite de fluxo excedido para oAWSconta”

Ajude-nos a melhorar esse tópico

Deixe-nos saber o que ajudaria a torná-lo melhor

Se você vir `"Error: You have exceeded the limit for the number of streams in your AWS account."`, poderá limpar os fluxos não utilizados em sua conta em vez de solicitar um aumento de limite.

Para limpar um fluxo não utilizado que você criou usando oAWS CLIou SDK:

```
aws iot delete-stream -stream-id value
```

Para obter mais detalhes, consulte delete-stream.

> **Note**
>
> Você pode usar `olist-streams`Comando para encontrar as IDs de fluxo.

Guia de solução de problemas do AWS IoT Device Defender

Ajude-nos a melhorar esse tópico

Deixe-nos saber o que ajudaria a torná-lo melhor

Geral

P: Há algum pré-requisito para usar oAWS IoT Device Defender?

A: Se você quiser usar as métricas relatadas por dispositivo, você deve primeiro implantar um agente no seuAWS IoTdispositivos conectados ou gateways de dispositivos. Os dispositivos devem fornecer um identificador de cliente consistente ou o nome da coisa.

Auditoria

P: Eu habilitei uma verificação e minha auditoria está exibindo “Em andamento” por muito tempo. Há algo de errado? Quando posso esperar resultados?

A: Quando a verificação é habilitada, a coleta de dados é iniciada imediatamente. No entanto, se sua conta tiver um grande volume de dados a serem coletados (por exemplo, certificados, coisas ou políticas), os resultados da verificação poderão não ficar disponíveis por algum tempo depois que você a habilitar.

Detectar

P: Como faço para saber quais limites definir em umAWS IoT Device Defendercomportamento do perfil de segurança do?

A: Comece criando um comportamento de perfil de segurança com limites baixos e anexe-os a um grupo de coisas contendo um conjunto representativo de dispositivos. Você pode usar o AWS IoT Device Defender para visualizar as métricas atuais e, em seguida, ajustar detalhadamente os limites de comportamento do dispositivo de acordo com o seu caso de uso.

P: Eu criei um comportamento, mas ele não está acionando uma violação quando eu espero. Como devo corrigir isso?

A: Quando você define um comportamento, você está especificando como espera que o dispositivo se comporte normalmente. Por exemplo, se você tiver uma câmera de segurança que apenas se conecta a um servidor central na porta TCP 8888, você não espera que ela faça nenhuma outra conexão. Para ser avisado se a câmera fizer uma conexão em outra porta, defina um comportamento como:

```
{
  "name": "Listening TCP Ports",
  "metric": "aws:listening-tcp-ports",
  "criteria": {
    "comparisonOperator": "in-port-set",
    "value": {
      "ports": [ 8888 ]
    }
  }
}
```

Se a câmera fizer uma conexão TCP na porta TCP 443, o comportamento do dispositivo será violado e acionará um alerta.

P: Um ou mais dos meus comportamentos estão em violação. Como faço para limpar a violação?

A: Os alarmes são limpos assim que o dispositivo retornar para o comportamento esperado, conforme definido pelos perfis de comportamento. Os perfis de comportamento são avaliados após o recebimento de dados de métricas para o dispositivo. Se o dispositivo não publicar nenhuma métrica por mais de dois dias, o evento de violação será definido automaticamente como `alarm-invalidated`.

P: Eu excluí um comportamento que estava em violação, mas como faço para interromper os alertas?

A: Excluir um comportamento interrompe todas as violações future e alertas para esse comportamento. Alertas anteriores devem ser drenados do mecanismo de notificação. Quando você exclui um comportamento, o registro de violações desse comportamento é retido durante o mesmo período como todas as outras violações na sua conta.

Métricas do dispositivo

P: Estou enviando relatórios de métricas que sei que violam meus comportamentos, mas nenhuma violação está sendo acionada. O que está errado?

A: Verifique se os relatórios de métricas estão sendo aceitos assinando os seguintes tópicos MQTT:

```
$aws/things/THING_NAME/defender/metrics/FORMAT/rejected
$aws/things/THING_NAME/defender/metrics/FORMAT/accepted
```

`THING_NAME`é o nome da coisa que informa a métrica e`FORMAT`O é “JSON” ou “CBOR” dependendo do formato do relatório de métricas enviado pela coisa.

Depois da assinatura, você receberá mensagens sobre esses tópicos para cada relatório de métricas enviado. A mensagem `rejected` indica que houve um problema ao analisar o relatório de métricas. Uma mensagem de erro é incluída na carga da mensagem para ajudá-lo a corrigir os erros no seu relatório da métricas. Uma`accepted`A mensagem indica que o relatório de métricas foi analisado corretamente.

P: O que acontece se eu enviar uma métrica vazia no meu relatório de métricas?

A: Uma lista vazia de portas ou endereços IP é sempre considerada em conformidade com o comportamento correspondente. Se o comportamento correspondente estava em violação, a violação será apagada.

P: Por que os relatórios de métrica do dispositivo contêm mensagens para dispositivos que não estão noAWS IoTRegistro do?

Se você tiver um ou mais perfis de segurança anexados a todas as coisas ou a todas as coisas não registradas, o AWS IoT Device Defender incluirá métricas de coisas não registradas. Se você quiser excluir métricas de coisas não registradas, poderá anexar os perfis a todos os dispositivos registrados em vez de todos os dispositivos.

P: Não vejo mensagens de um ou mais dispositivos não registrados, embora tenha aplicado um perfil de segurança a todos os dispositivos não registrados ou a todos os dispositivos. Como posso corrigir isso?

Verifique se você está enviando um relatório de métricas bem formado usando um dos formatos compatíveis. Para obter mais informações, consulte Especificação da documentação de métricas do dispositivo (p. 1022). Verifique se os dispositivos não registrados estão usando um identificador de cliente ou um nome de coisa consistentes. Se o nome da coisa contiver caracteres de controle ou tiver mais de 128 bytes de caracteres codificados em UTF-8, as mensagens relatadas pelos dispositivos serão rejeitadas.

P: O que acontece se um dispositivo não registrado for adicionado ao registro ou um dispositivo registrado for cancelado?

A: Se um dispositivo for adicionado ou removido do registro:

- Você verá duas violações diferentes para o dispositivo (uma com o nome da coisa registrada, uma sob sua identidade não registrada) se continuar a publicar métricas para violações. As violações ativas da identidade antiga param de aparecer após dois dias, mas estão disponíveis no histórico de violações por até 14 dias.

P: Qual valor devo fornecer no campo ID de relatório do relatório de métricas do dispositivo?

A: Use um valor único para cada relatório de métricas, expresso como um inteiro positivo. Uma prática comum é usar um timestamp epoch Unix.

P: Devo criar uma conexão MQTT dedicada paraAWS IoT Device DefenderMétricas do?

A: Não é necessária uma conexão MQTT separada.

P: Qual ID de cliente devo usar ao conectar para publicar métricas do dispositivo?

Para dispositivos (coisas) que estão no registro da AWS IoT, use o nome de coisa registrado. Para dispositivos que não estão no registro da AWS IoT, use um identificador consistente quando você se conectar à AWS IoT. Essa prática ajuda a corresponder as violações ao nome da coisa.

P: Posso publicar métricas para um dispositivo com outro client ID?

É possível publicar métricas em nome de outra coisa. Você pode fazer isso publicando as métricas no tópico reservado do AWS IoT Device Defender desse dispositivo. Por exemplo, `Thing-1` gostaria de publicar métricas para si mesmo e também em nome de `Thing-2`. `Thing-1` coleta suas próprias métricas e as publica no tópico MQTT:

```
$aws/things/Thing-1/defender/metrics/json
```

`Thing-1` obtém métricas do `Thing-2` e publica essas métricas no tópico MQTT:

```
$aws/things/Thing-2/defender/metrics/json
```

P: Quantos perfis de segurança e comportamentos posso ter na minha conta?

A: ConsulteAWS IoT Device DefenderEndpoints e cotas do.

P: Qual a aparência de uma função de destino prototípica para um destino de alerta?

A: Um papel que permiteAWS IoT Device DefenderPara publicar alertas em um destino de alerta (tópico do SNS) exige duas coisas:

- Uma relação de confiança que especifica iot.amazonaws.com como a entidade confiável.
- Uma política anexada que concede à AWS IoT permissão para publicar em um tópico do SNS especificado. Por exemplo:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "sns:Publish",
            "Resource": "<sns-topic-arn>"
        }
    ]
}
```

- Se o tópico do SNS usado para publicar alertas for um tópico criptografado, juntamente com a permissão para publicar no tópico do SNS,AWS IoTdeve receber mais duas permissões. Por exemplo:

```
{
    "Version": "2012-10-17",
    "Statement": [
       {
           "Effect": "Allow",
           "Action": [
               "sns:Publish",
               "kms:Decrypt",
               "kms:GenerateDataKey"
           ],
           "Resource": "<sns-topic-arn>"
       }
    ]
}
```

P: Meu envio de relatório de métrica com um tipo de métrica personalizado`number`falha com a mensagem de erro`Malformed metrics report`. O que está errado?

A: O tipo`number`só usa um único valor de métrica como entrada, mas ao enviar o valor das métricas na DeviceMetrics relatório, ele deve ser passado como uma matriz com um único valor. Certifique-se de que você está enviando o valor da métrica como uma matriz.

Carga útil de erro:

```
{"header":{"report_id":12334567,"version":"1.0"},"metrics":{"network_stats":
{"bytes_in":30680,"bytes_out":10652,"packets_in":113,"packets_out":118}},"custom_metrics":
{"my_custom_metric":{"number":0}}}
```

A mensagem de erro:

```
{"thingName":"myThing","status":"REJECTED","statusDetails":
{"ErrorCode":"InvalidPayload","ErrorMessage":"Malformed metrics
 report"},"timestamp":1635802047699}
```

Carga útil sem erro:

```
{"header":{"report_id":12334567,"version":"1.0"},"metrics":{"network_stats":
{"bytes_in":30680,"bytes_out":10652,"packets_in":113,"packets_out":118}},"custom_metrics":
{"my_custom_metric":[{"number":0}]}}
```

Resposta:

```
{"thingName":"myThing","12334567":1635800375,"status":"ACCEPTED","timestamp":1635801636023}
```

AWS IoTGuia de solução de problemas do Device Advisor

Ajude-nos a melhorar esse tópico

Deixe-nos saber o que ajudaria a torná-lo melhor

Geral

P: Posso executar vários conjuntos de testes em parallel?

A: Sim. O Device Advisor agora oferece suporte à execução de vários conjuntos de testes em diferentes dispositivos usando um endpoint no nível do dispositivo. Se você usar o endpoint no nível da conta, poderá executar um conjunto por vez porque um endpoint do Device Advisor está disponível por conta. Para obter mais informações, consulteConfigurar o dispositivo.

P: Vi pelo meu dispositivo que a conexão TLS foi negada pelo Device Advisor. Isso é esperado?

A: Sim. O Device Advisor nega a conexão TLS antes e depois de cada execução de teste. Recomendamos que os usuários implementem um mecanismo de nova tentativa de dispositivo para ter uma experiência de teste totalmente automatizada com o Device Advisor. Se você executar um conjunto de testes com mais de um caso de teste, por exemplo, conexão TLS, MQTT connect e MQTT publish, recomendamos que você tenha um mecanismo criado para o seu dispositivo. O mecanismo pode tentar se conectar ao nosso endpoint de teste a cada 5 segundos por um minuto a dois. Dessa forma, você pode executar vários casos de teste em sequência de forma automatizada.

P: Posso obter um histórico de chamadas de API do Device Advisor feitas na minha conta para fins de análise de segurança ou solução de problemas operacionais?

A: Sim. Para receber um histórico de chamadas de API do Device Advisor feitas em sua conta, basta ativar CloudTrail noAWS IoTManagement Console e filtre a origem do evento a se`iotdeviceadvisor.amazonaws.com`.

P: Como faço para visualizar os logs do Device Advisor no CloudWatch?

A: Os logs gerados durante a execução de um conjunto de testes são carregados para CloudWatch se você adicionar a política necessária (por exemplo,CloudWatchFullAccess) para sua função de serviço (consulteConfiguração (p. 1067)). Se houver pelo menos um caso de teste no conjunto de testes, um grupo de logs “AWS/IoT/DeviceAdvisor/$testSuiteID” será criado com dois fluxos de log. Um fluxo é o “$tesRunid” e inclui registros de ações realizadas antes e depois da execução dos casos de teste em seu conjunto de testes, como etapas de configuração e limpeza. O outro fluxo de log é “$SuiterUnid_$tesRunid”, que é específico para uma execução de suíte de testes. Eventos enviados de dispositivos eAWS IoT Coreserá registrado neste fluxo de log.

P: Qual é o propósito da função de permissão do dispositivo?

A: O Device Advisor fica entre o dispositivo de teste eAWS IoT Corepara simular cenários de teste. Ele aceita conexões e mensagens de seus dispositivos de teste e as encaminha paraAWS IoT Coreassumindo a função de permissão do dispositivo e iniciando uma conexão em seu nome. É importante garantir que as permissões de função do dispositivo sejam as mesmas do certificado que você usa para executar testes.AWS IoTpolíticas de certificado não são aplicadas quando o Device Advisor inicia uma conexão comAWS IoT Coreem seu nome usando a função de permissão do dispositivo. No entanto, as permissões da função de permissão do dispositivo que você definiu são aplicadas.

P: Em quais regiões o Device Advisor é suportado?

A: O Device Advisor é compatível com as regiões us-east-1, us-west-2, ap-northeast-1 e eu-west-1.

P: Por que vejo resultados inconsistentes?

A: Uma das principais causas de resultados inconsistentes é definir um teste`EXECUTION_TIMEOUT`Para um valor muito baixo. Para obter mais informações sobre recomendado e padrão`EXECUTION_TIMEOUT`Valores, consultecasos de teste Device Advisor.

P: Qual protocolo MQTT o Device Advisor suporta?

A: O Device Advisor suporta MQTT com certificados de cliente X509.

P: E se meu caso de teste falhar com uma mensagem de tempo limite de execução, mesmo que eu tentasse conectar meu dispositivo ao endpoint de teste?

A: Valide todas as etapas emCrie uma função do IAM para ser usada como função de dispositivo. Se o teste ainda falhar, pode ser que o dispositivo não esteja enviando a extensão SNI (Server Name Indication) correta, que é necessária para que o Device Advisor funcione. O valor SNI correto é o endereço do endpoint retornado ao seguir oConfigurar a seção de dispositivos.AWS IoTTambém exige que os dispositivos enviem a extensão SNI (Server Name Indication, indicação de nome de servidor) para o protocolo TLS (Transport Layer Security). Para obter mais informações, consulteSegurança de transporte noAWS IoT.

Solução de problemas de desconexões de frota de dispositivos

Ajude-nos a melhorar esse tópico

Deixe-nos saber o que ajudaria a torná-lo melhor

AWS IoTdesconexões da frota de dispositivos podem ocorrer por vários motivos. Este artigo explica como diagnosticar um motivo de desconexão e como lidar com desconexões causadas pela manutenção regular deAWS IoTserviço ou um limite de limitação.

Para diagnosticar o motivo da desconexão

Você pode verificar oAWSIOTlogsV2grupo de logs noCloudWatchpara identificar o motivo da desconexão no`disconnectReason`campo da entrada de log.

Você também pode usarAWS IoT'sEventos de ciclo de vidarecurso para identificar o motivo da desconexão. Se você se inscreveu noevento de desconexão do ciclo de vida(`$aws/events/presence/disconnected/clientId`), você receberá uma notificação deAWS IoTquando a desconexão acontece. Você pode identificar o motivo da desconexão no`disconnectReason`campo da notificação.

Para obter mais informações, consulteCloudWatchAWS IoTEntradas de log do doeEventos de ciclo de vida.

Para solucionar problemas de desconexões devido aAWS IoTmanutenção de serviços

Desconexões causadas porAWS IoTa manutenção do serviço é registrada como`SERVER_INITIATED_DISCONNECT`emAWS IoTevento de ciclo de vida e CloudWatch. Para lidar com essas desconexões, ajuste a configuração do lado do cliente para garantir que seus dispositivos possam ser reconectados automaticamente aoAWS IoTPlataforma.

Para solucionar problemas de desconexões devido a um limite de limitação

As desconexões causadas por um limite de limitação são registradas como`THROTTLED`emAWS IoTevento de ciclo de vida e CloudWatch. Para lidar com essas desconexões, você pode solicitaraumenta o limite do corretor de mensagensà medida que a contagem de dispositivos cresce.

Para obter mais informações, consulteAWS IoTOperador de mensagens principal.

Erros do AWS IoT

Ajude-nos a melhorar esse tópico

Deixe-nos saber o que ajudaria a torná-lo melhor

Esta seção lista os códigos de erro enviados pela AWS IoT.

Códigos de erro do agente de mensagens

Código de erro	Descrição do erro
400	Solicitação inválida.
401	Não autorizado.
403	Proibido.
503	Serviço indisponível.

Códigos de erro de segurança e identidade

Código de erro	Descrição do erro
401	Não autorizado.

Códigos de erro do Device Shadow

Código de erro	Descrição do erro
400	Solicitação inválida.
401	Não autorizado.
403	Proibido.
404	Não encontrado.
409	Conflito.
413	Solicitação muito grande.
422	Falha ao processar a solicitação.
429	Muitas solicitações.
500	Erro interno.
503	Serviço indisponível.

AWS IoTCotas do

Encontre informações sobreAWS IoTcotas noAWSReferência geral.

- para oAWS IoT Coreinformações sobre cotas, consulteAWS IoT CoreEndpoints e cotas do.
- para oAWS IoT Device Managementinformações sobre cotas, consulteAWS IoT Device ManagementEndpoints e cotas do.
- para oAWS IoT Device Defenderinformações sobre cotas, consulteAWS IoT Device DefenderEndpoints e cotas do.

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.