
AWS IoT Core

Guía para desarrolladores



AWS IoT Core: Guía para desarrolladores

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

Las marcas comerciales y la imagen comercial de Amazon no se pueden utilizar en relación con ningún producto o servicio que no sea de Amazon de ninguna manera que pueda causar confusión entre los clientes y que menoscalice o desacredite a Amazon. Todas las demás marcas comerciales que no sean propiedad de Amazon son propiedad de sus respectivos propietarios, que pueden o no estar afiliados, conectados o patrocinados por Amazon.

Table of Contents

¿Qué es AWS IoT?	1
Cómo acceden tus dispositivos y aplicacionesAWS IoT	1
¿Qué?AWS IoTpuede hacer	2
IoT en la industria	2
IoT en domótica	3
Cómo funciona AWS IoT	3
El universo de IoT	3
AWS IoTInformación general de servicios	6
Servicios de AWS IoT Core	8
Obtener más información sobre AWS IoT	11
Recursos de formación paraAWS IoT	11
AWS IoTRecursos y guías	12
AWS IoTTen redes sociales	12
AWSservicios utilizados por elAWS IoT Coremotor de reglas	13
Protocolos de comunicación compatibles conAWS IoT Core	14
Novedades deAWS IoTconsola	14
Leyenda	16
Introducción a AWS IoT Core	17
Connect del primer dispositivo aAWS IoT Core	17
Configurar suCuenta de AWS	18
Registro en una Cuenta de AWS	18
Cree un usuario y concédale permisos	19
Abra el iconoAWS IoTconsola	20
Pruebe el iconoAWS IoT Coretutorial interactivo	20
Conexión de dispositivos IoT	21
Guardar el estado del dispositivo sin conexión	22
Enrutamiento de datos de dispositivos a servicios	22
Pruebe el iconoAWS IoTQuick Connect	23
Paso 1. Inicie el tutorial	24
Paso 2. Crear objeto objeto objeto	26
Paso 3. Descarga archivos en tu dispositivo	28
Paso 4: Ejecución de la muestra	30
Paso 5. Exploración más	32
Comprobación de la conectividad con el punto final de datos de su dispositivo	33
ExplorarAWS IoT Coresservicios en tutorial práctico	37
¿Qué opción de dispositivo es la mejor para ti?	38
CrearAWS IoTRecursos	38
Configuración del dispositivo	42
Ver los mensajes MQTT con el cliente MQTT de AWS IoT	65
Visualización de mensajes MQTT en el cliente MQTT	66
Publicación de mensajes MQTT desde el cliente MQTT	67
Conexión a AWS IoT Core	69
AWS IoT Core- Puntos de enlace del plano de control	69
AWS IoTendpoints de dispositivo	70
AWS IoT Corepara puertas de enlace y dispositivos LoRaWAN	71
Conexión aAWS IoT CorePuntos de enlace de los servicios de	72
AWS CLI para AWS IoT Core	72
AWS SDK	73
AWS Mobile SDK	76
API de REST deAWS IoT CoreServicios de	77
Conexión de dispositivos aAWS IoT	78
AWS IoTdatos de dispositivos y puntos finales de servicio	78
SDK de dispositivos de AWS IoT	80
Protocolos de comunicación de dispositivos	81

Temas MQTT	98
Puntos de enlace configurables	116
Conexión aAWS IoT	
Puntos de enlace FIPS	122
AWS IoT Core- Puntos de enlace del plano de control	122
AWS IoT Core- Puntos de enlace del plano de datos	122
AWS IoT Device Management- puntos finales de datos de trabajos	123
AWS IoT Device Management- Puntos finales de Fleet Hub	123
AWS IoT Device Management- puntos de enlace de túnel seguros	123
Tutoriales de AWS IoT	124
Creación de demostraciones con elAWS IoTCliente del dispositivo	124
Requisitos previos para crear demostraciones con elAWS IoTCliente del dispositivo	125
Preparación de los dispositivos para elAWS IoTCliente del dispositivo	127
Instalar y configurar elAWS IoTCliente del dispositivo	137
Demostrar la comunicación de mensajes MQTT con elAWS IoTCliente del dispositivo	146
Demostrar acciones remotas (trabajos) con elAWS IoTCliente del dispositivo	159
Limpieza	169
Creación de soluciones con elAWS IoTSDKs de dispositivos	176
Comience a crear soluciones con elAWS IoTSDKs de dispositivos	176
Conexión de un dispositivo aAWS IoT Coremediante el uso de laAWS IoTSDK de dispositivos	177
CrearAWS IoTReglas para enrutar datos de dispositivos a otros servicios	193
Conservación del estado del dispositivo mientras el dispositivo está desconectado con Device Shadows	223
Creación de un autorizador personalizado paraAWS IoT Core	244
Control de la humedad del suelo conAWS IoTy Raspberry Pi	256
Administración de dispositivos con AWS IoT	266
Cómo administrar objetos con el registro	266
Creación de un objeto	267
Lista de objetos	267
Describe objetos	269
Actualización de un objeto	269
Eliminación de un objeto	269
Asociar un principal a un objeto	269
Desvincular un principal de un objeto	270
Tipos de objeto	270
Creación de un tipo de objeto	270
Lista de los tipos de objeto	271
Descripción de un tipo de objeto	271
Asociación de un tipo de objeto a un objeto	271
Descartar un tipo de objeto	272
Eliminación de un tipo de objeto	273
Grupos de objetos estáticos	273
Crear un grupo de objetos estático	274
Descripción de un grupo de objetos	275
Aregar un objeto a un grupo de objetos estático	275
Eliminar un objeto de un grupo de objetos estático	276
Enumarar los objetos en un grupo de objetos	276
Enumeración de grupos de objetos	276
Enumarar grupos para un objeto	278
Actualizar un grupo de objetos estático	278
Eliminación de un grupo de objetos	279
Asociar una política a un grupo de objetos estático	279
Desconectar una política de un grupo de objetos estático	280
Mostrar las políticas asociadas a un grupo de objetos estático	280
Enumaración de los grupos para una política	280
Obtención de políticas en vigor para un objeto	281
Prueba de autorización para acciones de MQTT	281
Grupos de objetos dinámicos	282

Crear un grupo de objetos dinámico	283
Describir un grupo de objetos dinámico	284
Actualizar un grupo de objetos dinámico	285
Eliminar un grupo de objetos dinámico	285
Limitaciones y conflictos	285
Etiquetado de los recursos de AWS IoT	288
Conceptos básicos de etiquetas	288
Restricciones y limitaciones en las etiquetas	289
Uso de etiquetas con políticas de IAM	289
Grupos de facturación	291
Visualización de datos de uso y asignación de costos	292
Seguridad	293
Seguridad en AWS IoT	293
Autenticación	294
AWS Training and Certification	294
Información general del certificado X.509	295
Autenticación del servidor	295
Autenticación del cliente	297
Autenticación personalizada	319
Autorización	331
AWS Training and Certification	333
Políticas de AWS IoT Core	333
Autorización de llamadas directas aAWSservicios de utilizandoAWS IoT Coreproveedor de credenciales	375
Acceso entre cuentas con IAM	379
Protección de los datos	381
Cifrado de datos en AWS IoT	382
Seguridad de transporte en AWS IoT	382
Cifrado de datos	383
Identity and Access Management	384
Público	384
Autenticación con identidades de IAM	385
Administración de acceso mediante políticas	387
Cómo AWS IoT funciona con IAM	389
Ejemplos de políticas basadas en identidad	408
Solución de problemas	411
Registro y monitorización	413
Herramientas de monitorización	413
Validación de conformidad	415
Resiliencia	415
Uso de AWS IoT Core conPuntos de conexión de la VPC	416
Creación de puntos de enlace de la VPC para AWS IoT Core	416
Control del acceso aAWS IoT Coresobre puntos de conexión de la VPC	417
Limitaciones de los extremos de la VPC	418
Escalado de puntos finales de VPC con IoT Core	418
Uso de dominios personalizados con puntos de enlace de la VPC	418
Disponibilidad de los puntos de enlace de la VPC paraAWS IoT Core	418
Seguridad de infraestructuras	419
Supervisión de seguridad	419
Prácticas recomendadas de seguridad	419
Protección de conexiones MQTT en AWS IoT	419
Mantener sincronizado el reloj del dispositivo	421
Validar el certificado de servidor	422
Usar una identidad única por dispositivo	422
Utilice un segundoRegión de AWScomo backup	423
Usar aprovisionamiento justo a tiempo	423
Permisos para ejecutarAWS IoTPrueba de Device Advisor	423

Prevención del suplente confuso entre servicios para Device Advisor	424
AWS Training and Certification	425
Monitorización de AWS IoT	426
Configuración de registros de AWS IoT	427
Configuración del rol y la política de registro	427
Configure el registro predeterminado en AWS IoT (consola)	428
Configurar el registro predeterminado en AWS IoT (CLI)	429
Configurar el inicio de sesión específico de recursos en AWS IoT (CLI)	431
Niveles de registro	432
MonitorAWS IoTalarmas y métricas mediante Amazon CloudWatch	433
Uso de las métricas de AWS IoT	433
Crear CloudWatch alarmas enAWS IoT	434
Métricas y dimensiones de AWS IoT	437
MonitorAWS IoTcon CloudWatch Registros	449
Visualización deAWS IoTinicios de sesión en el CloudWatch consola	449
CloudWatchAWS IoTEntradas de registro de en	449
Registre las llamadas a la API de AWS IoT con AWS CloudTrail	469
Información de AWS IoT en CloudTrail	470
Descripción de las entradas de los archivos de registro de AWS IoT	470
Reglas	472
Concesión de unAWS IoTregla el acceso que requiere	473
Transmisión de los permisos de rol	474
Creación de una regla de AWS IoT	475
Visualización de las reglas	479
Eliminar una regla	479
Acciones de reglas de AWS IoT	479
Apache Kafka	481
Alarmas de CloudWatch	489
CloudWatch Logs	490
Métricas de CloudWatch	491
DynamoDB	493
DynamoDBv2	495
Elasticsearch	497
HTTP	498
IoT Analytics	525
IoT Events	527
IoT SiteWise	528
Kinesis Data Firehose	532
Kinesis Data Streams	534
Lambda	536
OpenSearch	538
Republish	540
S3	541
Salesforce IoT IoT	543
SNS	543
SQS	545
Step Functions	547
Timestream	548
Solución de problemas de las reglas	553
Acceso a recursos entre cuentas medianteAWS IoTReglas de	553
Requisitos previos	553
Configuración multicuenta para Amazon SQS	553
Configuración entre cuentas para Amazon SNS	555
Configuración entre cuentas para Amazon S3	556
Configuración entre cuentas paraAWS Lambda	557
Control de errores (acción de error)	559
Formato de mensaje de acción de error	559

Ejemplo de acción de error	560
Reducción de costos de mensajería con Basic Ingest	560
Uso de Basic Ingest	561
Referencia de la SQL de AWS IoT	562
Cláusula SELECT	563
Cláusula FROM	564
Cláusula WHERE	565
Tipos de datos	565
Operadores	569
Funciones	575
Literales	619
Instrucciones case	619
Extensiones JSON	620
Plantillas de sustitución	622
Consultas de objetos anidados	623
Cargas binarias	624
Versiones de SQL	625
Servicio Device Shadow	627
Uso de sombras	627
Elegir utilizar sombras con nombre o sin nombre	627
Acceso a sombras	628
Uso de sombras en dispositivos, aplicaciones y otros servicios en la nube	628
Orden de los mensajes	629
Recorte de mensajes de sombra	630
Uso de sombras en dispositivos	631
Inicialización del dispositivo en la primera conexión a AWS IoT	632
Procesamiento de mensajes mientras el dispositivo está conectado a AWS IoT	633
Procesamiento de mensajes cuando el dispositivo se vuelve a conectar a AWS IoT	634
Uso de sombras en aplicaciones y servicios	634
Inicialización de la aplicación o el servicio al conectarse a AWS IoT	635
Procesamiento de cambios de estado mientras la aplicación o el servicio está conectado a AWS IoT	635
Detección de un dispositivo conectado	635
Simulación de comunicaciones del servicio Device Shadow	636
Configuración de la simulación	637
Inicializar el dispositivo	637
Enviar una actualización desde la aplicación	640
Responder a la actualización en el dispositivo	641
Observe la actualización en la aplicación	645
Más allá de la simulación	646
Interacción con sombras	646
Compatibilidad del protocolo	647
Estado de solicitud y notificación	647
Actualización de la sombra	647
Recuperación de un documento de sombra	650
Eliminación de datos de sombra	651
API REST de sombra de dispositivo	653
GetThingShadow	654
UpdateThingShadow	654
DeleteThingShadow	655
ListNamedShadowsForThing	656
Temas MQTT de sombra de dispositivo	657
/get	658
/get/accepted	658
/get/rejected	659
/update	660
/update/delta	661

/update/accepted	661
/update/documents	662
/update/rejected	663
/delete	663
/delete/accepted	664
/delete/rejected	665
Documentos del servicio Device Shadow	665
Ejemplos de documento de sombra	665
Propiedades del documento	670
Estado delta	671
Control de versiones de documentos de sombra	672
Tokens de cliente en documentos de sombra	672
Propiedades de documento de sombra vacío	672
Valores de matriz en documentos sombra	673
Mensajes de error de Device Shadow	674
Trabajos	675
Conceptos clave de trabajos	675
Administrar trabajos	677
Firma de código para trabajos	678
documento de Job	678
URL prefirmadas	678
Creación y administración de trabajos con la consola	679
Creación y administración de trabajos mediante la CLI	680
Plantillas de trabajo	688
Personalizado y AWSplantillas administradas	688
UsarAWSplantillas administradas	689
Creación de plantillas de trabajo personalizadas	702
Trabajoconfigurations	707
Cómo funcionan las configuraciones de trabajos	708
Especifica configuraciones adicionales	713
Dispositivos y trabajos	718
Programación de dispositivos para trabajar con trabajos	720
Flujo de trabajo del dispositivo	720
Flujo de trabajo	721
Notificaciones de trabajos	724
AWS IoTAPI de trabajos	729
API de administración y control de trabajo y tipos de datos	731
API de MQTT y HTTPS de dispositivo de trabajo y tipos de datos	745
Protección de usuarios y dispositivos para trabajos	755
Tipo de política de obligatorioAWS IoTTrabajos	755
Autorización de usuarios de trabajos y servicios en la nube	756
Autorización de dispositivos para utilizar trabajos	762
Límites de los trabajos	765
Tunelización segura de AWS IoT	766
¿Qué es el túnel seguro?	766
Conceptos de tunelización segura	766
Cómo funciona el tunelización segura de	767
Ciclo de vida del túnel seguro	768
AWS IoTTutoriales de tunelización segura	768
Abra un túnel e inicie la sesión SSH en un dispositivo remoto	768
Proxy local	771
Cómo utilizar el proxy local	771
Configurar proxy local para dispositivos que utilizan proxy web	775
Flujos de datos multiplex en un túnel seguro	780
Fragmento de agente de IoT	784
Control del acceso a los túneles	785
Requisitos previos de acceso al túnel	785

Políticas de acceso a tú	785
Configuración de un dispositivo remoto	790
Resolución de problemas de conectividad de túneles seguros	791
Error de token de acceso de cliente no válido	791
Error de discrepancia de token de cliente	791
Problemas de conectividad de dispositivos	792
Aprovisionamiento de dispositivos	795
Dispositivos de aprovisionamiento en AWS IoT	796
API de aprovisionamiento de flotas	796
Aprovisionamiento de dispositivos que no tienen certificados de dispositivo mediante el aprovisionamiento de flotas	797
Aprovisionamiento por reclamación	797
Aprovisionamiento por usuario de confianza	799
Uso de enlaces de preaproxionamiento con la CLI de AWS	801
Aprovisionamiento de dispositivos que tienen certificados de dispositivo	803
Aprovisionamiento de un solo objeto	803
Aprovisionamiento justo a tiempo	804
Registro masivo	807
Aprovisionamiento de plantillas	808
Sección de parámetros	808
Sección de recursos	808
Ejemplo de plantilla para el registro JITP y masivo	812
Aprovisionamiento de flotas	814
Enlaces de preaproxionamiento	816
Preaproxionamiento de entrada de enlace	817
Valor de retorno del enlace previo a la provisión	817
Ejemplo de Lambda de enlace de preaproxionamiento	817
API de MQTT de aprovisionamiento de dispositivos	819
CreateCertificateFromCsr	819
CreateKeysAndCertificate	821
RegisterThing	823
Indexación de flotas	825
Administración de la indexación del	826
Indexación de elementos	826
Indexación de grupos de elementos	826
Campos administrados	827
Campos personalizados	828
Administración de la indexación de objetos	829
Administración de la indexación de grupos de objetos	839
Consulta de datos agregados	840
GetStatistics	840
GetCardinality	842
GetPercentiles	843
Agregación GetBuckets	845
Autorización	845
Sintaxis de la consulta	846
Características admitidas	846
Características de no admitidas	846
Notas	846
Ejemplo de consultas de objetos	847
Ejemplo de consultas de grupo de objetos	850
Métricas de flota	851
Explicación introductoria	851
Administración de métricas de flota	856
Entrega de archivos basada en MQTT	861
¿Qué es un flujo?	861
Administración de una transmisión en el AWS Nube	862

Conceder permisos a tus dispositivos	862
Connect los dispositivos a AWS IoT	863
Uso de AWS IoTEntrega de archivos basada en MQTT en dispositivos	863
Utilizar DescribeStream para obtener datos de transmisión	863
Obtener bloques de datos de un archivo de transmisión	865
Gestión de errores de AWS IoTEntrega de archivos basada en MQTT	869
Ejemplo de caso de uso en FreeRTOS OTA	870
AWS IoT Device Defender	871
AWS Training and Certification	871
Introducción a AWS IoT Device Defender	871
Configuración	871
Guía de auditoría	873
Guía de detección de ML	885
Personaliza cuándo y cómo lo ves AWS IoT Device Defender resultados de auditoría	908
Auditoría	919
Gravedad del problema	919
Pasos siguientes	920
Comprobaciones de auditoría	920
Comandos de auditoría	947
Supresiones de hallazgos de auditoría	974
Detect	983
Monitorización del comportamiento de dispositivos no registrados	985
Casos de uso de seguridad	985
Conceptos	990
Comportamientos	991
Detección de ML	993
Métricas de personalizadas	997
Métricas del lado del dispositivo	1003
Métricas del lado de la nube	1019
Establecer el ámbito de las métricas en los perfiles de seguridad utilizando dimensiones	1026
Permisos	1032
Comandos de detección	1033
Cómo utilizar AWS IoT Device Defender Detect	1035
Acciones de mitigación	1036
Acciones de mitigación de auditoría	1037
Detección de acciones de mitigación	1039
Cómo definir y administrar las acciones de mitigación	1039
Aplicación de acciones de mitigación	1044
Permisos	1049
Comandos de las acciones de mitigación	1052
Uso de AWS IoT Device Defender con otros servicios de AWS	1053
Uso de AWS IoT Device Defender con dispositivos en ejecución AWS IoT Greengrass	1053
Uso de AWS IoT Device Defender con FreeRTOS y dispositivos integrados	1053
Uso de AWS IoT Device Defender con AWS IoT Device Management	1054
Prevención del suplente confuso entre servicios	1054
Prácticas recomendadas de seguridad para agentes de dispositivos	1055
Asesor de dispositivos	1058
Configuración	1058
Cree una objeto de IoT	1059
Crear un rol de IAM para utilizarlo como rol de dispositivo	1059
Crear una política administrada personalizada para que un usuario de IAM utilice Device Advisor	1060
Crear un usuario de IAM para usar Device Advisor	1061
Configuración del dispositivo	1061
Introducción a Device Advisor en la consola	1062
Flujo de trabajo del Advisor	1068
Requisitos previos	1068
Creación de una definición de conjunto de pruebas	1068

Obtenga una definición de conjunto de pruebas	1070
Obtener un punto final de prueba	1070
Iniciar una ejecución de un conjunto de pruebas	1071
Ejecutar un conjunto de pruebas	1071
Detener la ejecución de un conjunto de pruebas	1071
Obtenga un informe de calificación para que se ejecute correctamente el conjunto de pruebas de calificación	1072
Flujo de trabajo de consola detallado de Device	1072
Requisitos previos	1073
Creación de una definición de conjunto de pruebas	1073
Iniciar una ejecución de un conjunto de pruebas	1077
Detener la ejecución de un conjunto de pruebas (opcional)	1079
Ver detalles y registros de ejecución del conjunto de pruebas	1081
Descarga de AWS IoT Informe de calificación	1082
Casos de prueba de Device Advisor	1082
TLS	1082
Permisos y políticas	1086
MQTT	1087
Sombra	1094
Ejecución de Job	1095
Mensajes de los eventos	1097
Cómo se generan los mensajes de eventos	1097
Política de recepción de mensajes de eventos	1097
Habilitar eventos para AWS IoT	1097
Eventos de registro	1100
Eventos de objeto	1101
Eventos de tipo de objeto	1102
Eventos de grupos de objetos	1104
Eventos de trabajos	1108
Eventos del ciclo de vida	1111
Eventos de conexión/desconexión	1111
Eventos de suscripción/cancelación de suscripción	1114
AWS IoT Corefor LoRaWAN	1116
Introducción	1116
Cómo utilizar AWS IoT Corefor LoRaWAN	1116
AWS IoT Corefor LoRaWAN Regiones y puntos de enlace	1117
AWS IoT Corefor LoRaWAN	1117
¿Qué es AWS IoT Corefor LoRaWAN?	1117
¿Qué es LoRaWAN?	1118
Cómo AWS IoT Corefor LoRaWAN	1119
Conexión de puertas de enlace y dispositivos a AWS IoT Core for LoRaWAN	1120
Convenciones de nomenclatura para sus dispositivos, puertas de enlace, perfiles y destinos	1120
Asignación de datos de dispositivo a datos de servicio	1120
Uso de la consola para incorporar el dispositivo y la puerta de enlace a AWS IoT Core for LoRaWAN	1121
Describa sus AWS IoT Core para LoRaWAN Recursos WAN	1121
Incorporación de sus gateways a AWS IoT Core for LoRaWAN	1123
Incorporación de sus dispositivos a AWS IoT Core for LoRaWAN	1129
Conexión a AWS IoT Core for LoRaWAN mediante un punto de enlace de interfaz de la VPC	1140
Incorporación AWS IoT Core for LoRaWAN Punto de enlace de API del plano de control	1142
Incorporación AWS IoT Core for LoRaWAN Puntos de enlace de API del plano de datos	1144
Administración de gateways con AWS IoT Core for LoRaWAN	1150
Requisito de software LoRa Basics Station	1151
Uso de puertas de enlace calificadas desde el AWS Catalogo de dispositivos asociados	1151
Uso de protocolos CUPS y LNS	1151
Configurar las subbandas y las capacidades de filtrado de la puerta de enlace	1151
Actualizar el firmware de la puerta de enlace mediante el servicio AWS IoT Core for LoRaWAN ...	1154

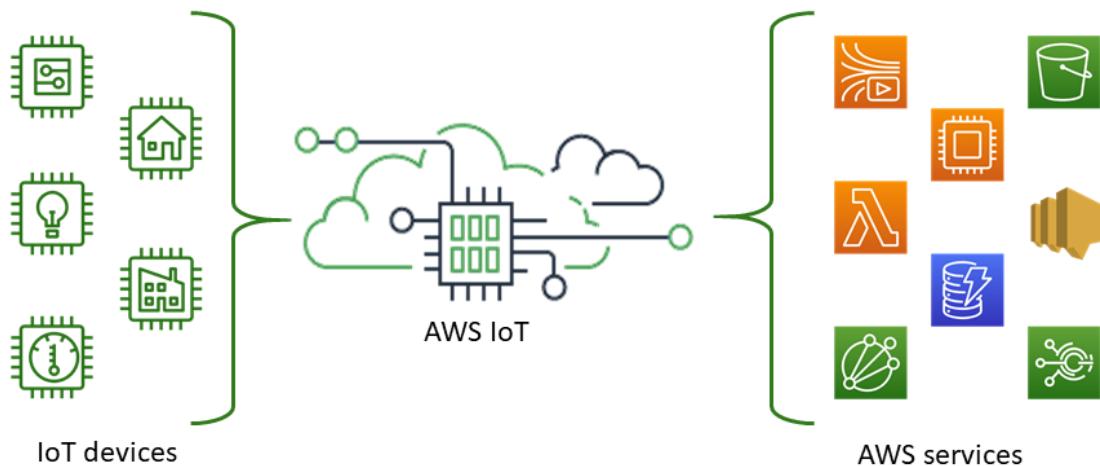
Administración de dispositivos con AWS IoT Core for LoRaWAN	1164
Consideraciones sobre el dispositivo	1164
Uso de dispositivos con puertas de enlace calificadas para AWS IoT Core for LoRaWAN	1164
LoRaVersión WAN	1165
Modos de activación	1165
Clases de dispositivo	1165
Gestiona la comunicación entre tuLoRaDispositivos WAN y AWS IoT	1165
Crear grupos de multidifusión para enviar una carga útil de enlace descendente a varios dispositivos	1172
Actualizaciones de firmware por aire (FUOTA) para AWS IoT Core for LoRaWAN Dispositivos de ..	1183
Supervisión de su flota de recursos inalámbricos en tiempo real mediante el analizador de redes	1193
Agregar el rol de IAM necesario para el analizador de redes	1194
Cree una configuración de analizador de red y agregue recursos	1196
Transmitir mensajes de seguimiento del analizador de red con WebSockets	1202
Ver y supervisar los registros de mensajes de seguimiento del analizador de red en tiempo real ..	1207
Seguridad de los datos con AWS IoT Core for LoRaWAN	1209
Cómo se protegen los datos en todo el sistema	1210
Seguridad del transporte de dispositivos y pasarela LoRaWAN	1210
Integración de Amazon Sidewalk para AWS IoT Core	1212
Cómo incorporar tu dispositivo Sidewalk	1212
Dispositivos Sidewalk integrados con integración de Amazon Sidewalk para AWS IoT Core	1212
Agregue las credenciales de su cuenta de Sidewalk	1213
Añadir un destino para su dispositivo Sidewalk	1214
Crear reglas para procesar mensajes de dispositivos Sidewalk	1216
Connect tu dispositivo Sidewalk y visualiza el formato de metadatos de enlace ascendente	1218
Connect el dispositivo Sidewalk	1218
Ver formato de los mensajes de enlace ascendente	1218
Monitoreo y registro de AWS IoT Wireless uso de Amazon CloudWatch	1220
Configuración del registro para AWS IoT Wireless	1221
Creación de una política y un rol de registro para AWS IoT Wireless	1221
Configuración de registro en AWS IoT Wireless recursos	1223
Monitor AWS IoT Wireless con CloudWatch Registro	1231
Vista CloudWatch AWS IoT Wireless Entradas de registro de en	1232
Usar CloudWatch Perspectivas para filtrar registros para AWS IoT Wireless	1238
Notificaciones de eventos de AWS IoT Wireless	1241
Cómo se puede notificar a los recursos de los eventos	1241
Eventos y tipos de recursos	1241
Política para recibir notificaciones de eventos inalámbricos	1242
Formato de los temas MQTT para eventos inalámbricos	1242
Precios de eventos inalámbricos	1244
Habilitar eventos para recursos inalámbricos	1244
Configuraciones de eventos	1244
Requisitos previos	1245
Habilitar notificaciones mediante el AWS Management Console	1245
Habilitar notificaciones mediante el AWS CLI	1246
Notificaciones de eventos para recursos de LoRaWAN	1247
Tipos de eventos para los recursos de LoRaWAN	1248
LoRaWAN	1248
Eventos de estado de conexión	1250
Notificaciones de eventos para recursos Sidewalk	1252
Tipos de eventos para recursos de Sidewalk	1252
Eventos de estado de registro de dispositivos	1252
Eventos de proximidad	1255
Integración de Alexa Voice Service (AVS) para AWS IoT	1257
Introducción a la integración de Alexa Voice Service (AVS) para AWS IoT con un dispositivo NXP	1258
Vista previa de la integración de Alexa Voice Service (AVS) para AWS IoT con una cuenta de NXP preconfigurada	1259

Interactúa con Alexa	1270
Utilizar su AWSy cuentas de desarrollador de Alexa Voice Service para configurar AVS para AWS IoT	1272
AWS IoTSDK de dispositivos, SDK móviles y AWS IoTCliente de dispositivos	1274
SDK de dispositivos de AWS IoT	1274
SDK de dispositivos de AWS IoT para Embedded C	1275
Antes AWS IoT Versiones del SDK de dispositivos	1276
AWS Mobile SDK	1276
AWS IoTCliente de dispositivos	1277
Solución de problemas	1278
Diagnóstico de problemas de conectividad	1278
Conexión	1278
Autenticación	1279
Autorización	1280
Seguridad e identidad	1280
Diagnóstico de problemas de las reglas	1281
Configuración de CloudWatch Logs para la solución de problemas	1281
Diagnóstico de servicios externos	1282
Diagnóstico de problemas de SQL	1282
Diagnóstico de problemas relacionados con las sombras	1282
Diagnosticar problemas con acciones de Salesforce	1283
Registro de seguimiento de ejecución	1283
Éxito y error de una acción	1284
Guía para solucionar problemas de indexación de flota	1284
Solución de problemas de consultas de agregación en el servicio de indexación de flotas	1284
Solución de problemas de métricas de flota	1285
Solución de problemas de «Superación del límite de secuencias de AWS cuenta»	1286
Guía para solucionar problemas de AWS IoT Device Defender	1286
AWS IoT Guía para solucionar problemas de Device Advisor	1290
Solución de problemas de desconexión de flota de dispositivos	1291
Errores de AWS IoT	1292
AWS IoT Cuotas de	1294
	mccxcv

¿Qué es AWS IoT?

AWS IoT proporciona los servicios en la nube que conectan los dispositivos IoT a otros dispositivos y servicios en la nube de AWS. AWS IoT proporciona software para dispositivos que puede ayudarlo a integrar los dispositivos IoT en soluciones basadas en AWS IoT. Si los dispositivos se pueden conectar a AWS IoT, AWS IoT puede conectarlos a los servicios en la nube que proporciona AWS.

Para obtener una introducción práctica a AWS IoT, visita [Introducción a AWS IoT Core \(p. 17\)](#).



AWS IoT te permite seleccionar el más adecuado y up-to-date tecnologías para su solución. Para ayudarte a administrar y dar soporte a sus dispositivos IoT sobre el terreno, AWS IoT Core admite estos protocolos:

- [MQTT \(Message Queue Server y transporte de telemetría\) \(p. 85\)](#)
- [MQTT sobre WSS \(Websockets Secure\) \(p. 85\)](#)
- [HTTPS \(protocolo de transferencia de hipertexto: seguro\) \(p. 95\)](#)
- [LoRaWAN \(red de área extendida de largo alcance\) \(p. 1116\)](#)

La AWS IoT Core admite dispositivos y clientes que utilizan MQTT y MQTT sobre protocolos WSS para publicar y suscribirse a mensajes. También admite dispositivos y clientes que utilizan el protocolo HTTPS para publicar mensajes.

AWS IoT Core para LoRaWAN le ayuda a conectarse y administrar la red inalámbrica LoRaWAN (red de área amplia de largo alcance de bajo consumo). AWS IoT Core para LoRaWAN reemplaza la necesidad de desarrollar y operar un LoRaWAN Servidor de red (LNS).

Si no lo necesitas, AWS IoT tiene características tales como comunicaciones de dispositivos, [Reglas de \(p. 472\)](#), o bien [jobs \(p. 675\)](#), consulta [AWS Mensajería](#) para obtener información sobre otros servicios AWS IoT de mensajería que podrían ajustarse mejor a sus necesidades.

Cómo acceden tus dispositivos y aplicacionesAWS IoT

AWS IoT proporciona las siguientes interfaces para [Tutoriales de AWS IoT \(p. 124\)](#):

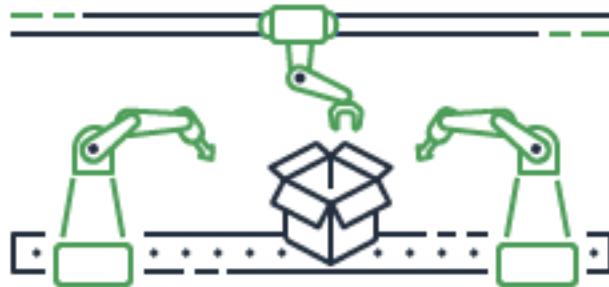
- AWS IoT SDKs de dispositivos: cree aplicaciones en sus dispositivos para enviar y recibir mensajes de AWS IoT. Para obtener más información, consulte [AWS IoT SDK de dispositivos, SDK móviles y AWS IoT Cliente de dispositivos \(p. 1274\)](#).
- AWS IoT Core para LoRaWAN—Conecte y administre su WAN de largo alcance (LoRa dispositivo y puertas de enlace (WAN) mediante [AWS IoT Core para LoRaWAN \(p. 1116\)](#).
- AWS Command Line Interface (AWS CLI): ejecuta comandos para AWS IoT en Windows, macOS X y Linux. Estos comandos le permiten crear y administrar objetos de objeto, certificados, reglas, trabajos y políticas. Para empezar, consulte la [AWS Command Line Interface Guía de usuario](#). Para obtener más información sobre los comandos de AWS IoT, consulte la [AWS CLI Reference](#) de los comandos.
- AWS IoT API - Cree sus aplicaciones IoT mediante solicitudes HTTP o HTTPS. Estas acciones de la API le permiten crear y administrar objetos de objeto, certificados, reglas y políticas mediante programación. Para obtener más información acerca de las acciones de la API para AWS IoT, consulte las [acciones](#) en las referencias de API de AWS IoT.
- AWSSDK de- Cree sus aplicaciones IoT mediante API específicas de un lenguaje. Estos SDK integran las API de HTTP/HTTPS y le permiten programar en cualquiera de los lenguajes admitidos. Para obtener más información, consulte [SDK y herramientas de AWS](#).

También puedes acceder a AWS IoT a través de [AWS IoT consola](#), que proporciona una interfaz gráfica de usuario (GUI) a través de la cual puede configurar y administrar objetos, certificados, reglas, trabajos, políticas y otros elementos de sus soluciones de IoT.

¿Qué?AWS IoT puede hacer

En este tema se describen algunas de las soluciones que podría necesitar AWS IoT admite.

IoT en la industria



Estos son algunos ejemplos de AWS IoT soluciones para [casos de uso industrial](#) que aplican tecnologías de IoT para mejorar el rendimiento y la productividad de los procesos industriales.

Soluciones para casos de uso industrial

- [Usar AWS IoT para crear modelos de calidad predictiva en operaciones industriales](#)

Vea cómo AWS IoT puede recopilar y analizar datos de operaciones industriales para crear modelos de calidad predictivos. [Más información](#)

- [Usar AWS IoT para apoyar el mantenimiento predictivo en operaciones industriales](#)

Vea cómo AWS IoT puede ayudar a planificar el mantenimiento preventivo para reducir el tiempo de inactividad no planificado. [Más información](#)

IoT en domótica



Estos son algunos ejemplos de AWS IoT soluciones para [Casos de uso de Domótica](#) que aplican tecnologías de IoT para crear aplicaciones de IoT escalables que automatizan las actividades domésticas mediante dispositivos domésticos conectados.

Soluciones para domótica

- [Usar AWS IoT en tu casa conectada](#)

Vea cómo AWS IoT puede proporcionar soluciones domóticas integradas.

- [Usar AWS IoT para proporcionar seguridad y monitoreo en el hogar](#)

Vea cómo AWS IoT puede aplicar machine learning y edge computing a su solución de domótica.

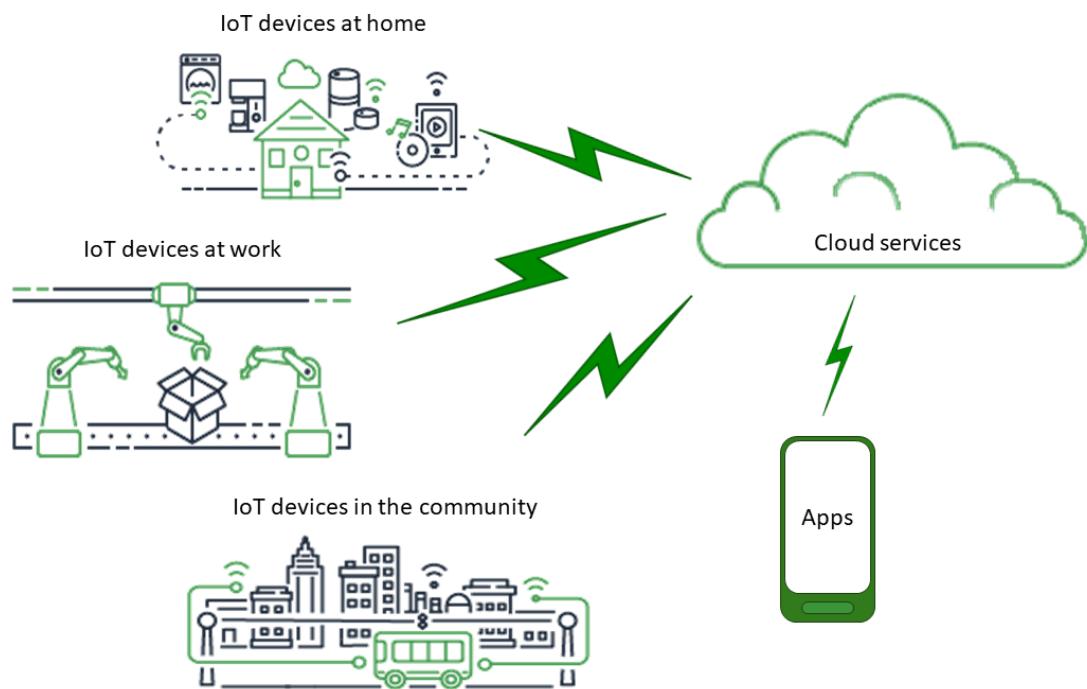
Para obtener una lista de soluciones para casos de uso industrial, de consumo y comerciales, consulte la [AWS IoT Repositorio de soluciones](#).

Cómo funciona AWS IoT

AWS IoT proporciona servicios en la nube y soporte de dispositivos que puede utilizar para implementar soluciones de IoT. AWS proporciona muchos servicios en la nube para admitir aplicaciones basadas en IoT. Para ayudarle a entender por dónde empezar, esta sección proporciona un diagrama y una definición de conceptos esenciales para presentarle el universo de IoT.

El universo de IoT

En general, Internet de las cosas (IoT) consiste en los componentes clave que se muestran en este diagrama.



móviles

Las aplicaciones proporcionan a los usuarios finales acceso a dispositivos IoT y a las funciones proporcionadas por los servicios en la nube a los que están conectados esos dispositivos.

Servicios en la nube

Los servicios en la nube son servicios distribuidos de almacenamiento y procesamiento de datos a gran escala que están conectados a Internet. Entre los ejemplos se incluyen:

- Servicios de administración y conexión de IoT.
AWS IoT es un ejemplo de un servicio de administración y conexión de IoT.
- Servicios informáticos, como Amazon Elastic Compute Cloud y AWS Lambda.
- Servicios de bases de datos, como Amazon DynamoDB

Comunicaciones

Los dispositivos se comunican con los servicios en la nube mediante diversas tecnologías y protocolos. Entre los ejemplos se incluyen:

- Wifi/Internet de banda ancha
- Datos móviles de banda ancha
- Datos móviles de banda estrecha
- Red de área extensa de largo alcance (LoRaWAN)
- Comunicaciones RF propietarias

Dispositivos

Un dispositivo es un tipo de hardware que administra interfaces y comunicaciones. Por lo general, los dispositivos se encuentran muy cerca de las interfaces del mundo real que monitorean y controlan. Los dispositivos pueden incluir recursos informáticos y de almacenamiento, como microcontroladores, CPU y memoria. Entre los ejemplos se incluyen:

- Raspberry Pi
- Arduino
- Asistentes de interfaz de voz
- LoRaWAN y dispositivos
- Dispositivos Amazon Sidewalk
- Dispositivos de IoT personalizados

Interfaces

Una interfaz es un componente que conecta un dispositivo al mundo físico.

- Interfaces de usuario

Componentes que permiten a los dispositivos y a los usuarios comunicarse entre sí.

- Interfaces de entrada

Permitir a un usuario comunicarse con un dispositivo

Ejemplos: teclado, botón

- Interfaces de salida

Habilitar que un dispositivo se comunique con un usuario

Ejemplos: Pantalla alfanumérica, pantalla gráfica, indicador luminoso, timbre de alarma

- Sensores

Componentes de entrada que miden o detectan algo en el mundo exterior de una forma que un dispositivo entiende. Entre los ejemplos se incluyen:

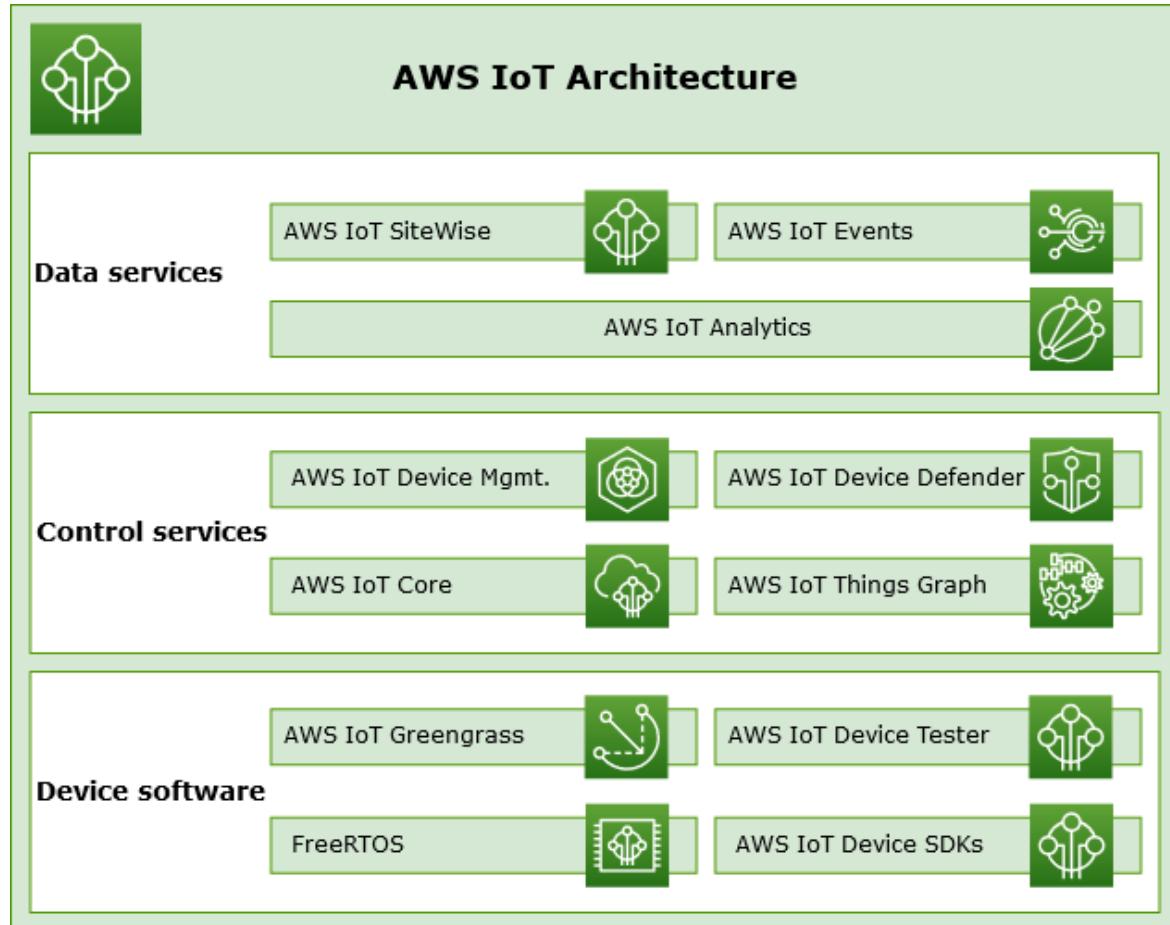
- Sensor de temperatura (convierte la temperatura en una señal analógica o digital)
- Sensor de humedad (convierte la humedad relativa en una señal analógica o digital)
- Convertidor analógico a digital (convierte un voltaje analógico en un valor numérico)
- Unidad de medición de distancia ultrasónica (convierte una distancia en un valor numérico)
- Sensor óptico (convierte un nivel de luz en un valor numérico)
- Cámara (convierte los datos de imagen en datos digitales)
- Actuadores

Componentes de salida que el dispositivo puede utilizar para controlar algo en el mundo exterior. Entre los ejemplos se incluyen:

- Motores paso a paso (convertir señales eléctricas en movimiento)
- Relés (controlan tensiones y corrientes eléctricas elevadas)

AWS IoT Información general de servicios

En el universo IoT, AWS IoT proporciona los servicios que admiten los dispositivos que interactúan con el mundo y los datos que pasan entre ellos y AWS IoT. AWS IoT se compone de los servicios que se muestran en esta ilustración para dar soporte a su solución IoT.



AWS IoT software de dispositivo

AWS IoT proporciona este software para admitir sus dispositivos IoT.

AWS IoT Greengrass

AWS IoT Greengrass extiende AWS IoT a dispositivos de borde de manera sencilla, de modo que puedan actuar a nivel local en función de los datos que generan y utilizar la nube para tareas de administración, análisis y almacenamiento duradero. Con AWS IoT Greengrass, los dispositivos conectados pueden funcionar AWS Lambda funciones, contenedores Docker o ambos, ejecutan predicciones basadas en modelos de aprendizaje automático, mantienen los datos de dispositivos sincronizados y se comunican con otros dispositivos de manera segura, incluso sin estar conectados a Internet.

AWS IoT Device Tester

AWS IoT Device Tester para FreeRTOS y AWS IoT Greengrass es una herramienta de automatización de pruebas para microcontroladores. AWS IoT Device Tester prueba su dispositivo para determinar si ejecutará FreeRTOS o AWS IoT Greengrass interoperar con AWS IoT Services de .

SDK de dispositivos de AWS IoT

La [AWS IoT SDK para dispositivos y móviles \(p. 1274\)](#) le ayudan a conectar eficazmente sus dispositivos a AWS IoT. Los de AWS IoT para dispositivos y móviles contienen bibliotecas de código abierto, guías para desarrolladores con ejemplos y guías de portabilidad para que pueda crear innovadores productos y soluciones de IoT en las plataformas de hardware que prefiera.

FreeRTOS

[FreeRTOS](#) es un sistema operativo de código abierto en tiempo real para microcontroladores que le permite incluir dispositivos perimetrales pequeños y de bajo consumo en su solución IoT. FreeRTOS incluye un núcleo y un conjunto creciente de bibliotecas de software que admiten muchas aplicaciones. Los sistemas FreeRTOS pueden conectar de forma segura sus dispositivos pequeños y de bajo consumo a [AWS IoT](#) y admite dispositivos perimetrales más potentes en ejecución [AWS IoT Greengrass](#).

AWS IoT CoreDevice Advisor

[AWS IoT CoreDevice Advisors](#) una capacidad de prueba totalmente administrada y basada en la nube para validar dispositivos IoT durante el desarrollo de software de dispositivos. Device Advisor proporciona pruebas predefinidas que puede utilizar para validar dispositivos IoT para lograr una conectividad fiable y segura con [AWS IoT Core](#), antes de desplegar dispositivos en producción.

AWS IoT servicios de control

Connect a lo siguiente [AWS IoT services](#) para administrar los dispositivos de su solución IoT.

AWS IoT Core

[AWS IoT Core](#) es un servicio en la nube administrado que permite a los dispositivos conectados interaccionar de forma segura con las aplicaciones en la nube y otros dispositivos. [AWS IoT Core](#) puede admitir muchos dispositivos y mensajes, y puede procesar y enrutar esos mensajes a [AWS IoT endpoints](#) y otros dispositivos. Con [AWS IoT Core](#), tus aplicaciones pueden interactuar con todos tus dispositivos incluso cuando no están conectados.

AWS IoT Administración de dispositivos

[AWS IoT Administración de dispositivos](#) los servicios le ayudan a rastrear, supervisar y administrar la gran cantidad de dispositivos conectados que componen sus flotas de dispositivos. [AWS IoT](#) Los servicios de administración de dispositivos le ayudan a garantizar que los dispositivos IoT funcionen correctamente y de forma segura después de implementarlos. También proporcionan túneles seguros para acceder a sus dispositivos, supervisar su estado, detectar y solucionar problemas de forma remota, así como servicios para administrar las actualizaciones de software y firmware del dispositivo.

AWS IoT Device Defender

[AWS IoT Device Defender](#) le ayuda a proteger su flota de dispositivos IoT. [AWS IoT Device Defender](#) audita continuamente las configuraciones de IoT para asegurarse de que no se desvian de las prácticas recomendadas de seguridad. [AWS IoT Device Defender](#) envía una alerta cuando detecta cualquier brecha en la configuración de IoT que podría crear un riesgo de seguridad, como los certificados de identidad que se comparten en varios dispositivos o un dispositivo con un certificado de identidad revocado que intenta conectarse a [AWS IoT Core](#).

AWS IoT Things Graph

[AWS IoT Things Graph](#) es un servicio que le permite conectar visualmente diferentes dispositivos y servicios web para crear aplicaciones de IoT. [AWS IoT Things Graph](#) proporciona un aspecto visual drag-and-drop interfaz para conectar y coordinar las interacciones entre dispositivos y servicios web, de modo que pueda crear aplicaciones de IoT de forma eficiente.

Servicios de datos de AWS IoT

Analice los datos de los dispositivos de su solución IoT y tome las medidas adecuadas utilizando lo siguiente: AWS IoT Servicios de .

AWS IoT Analytics

AWS IoT Analytics permite ejecutar y poner en funcionamiento análisis sofisticados de volúmenes masivos de datos de IoT no estructurados. AWS IoT Analytics automatiza cada paso difícil que se requiere para analizar datos desde dispositivos de IoT. AWS IoT Analytics filtra, transforma y enriquece datos de IoT antes de almacenarlos en un almacén de datos de series temporales para su análisis. Puede analizar los datos ejecutando consultas puntuales o programadas mediante el motor de consultas SQL integrado o el aprendizaje automático.

AWS IoT SiteWise

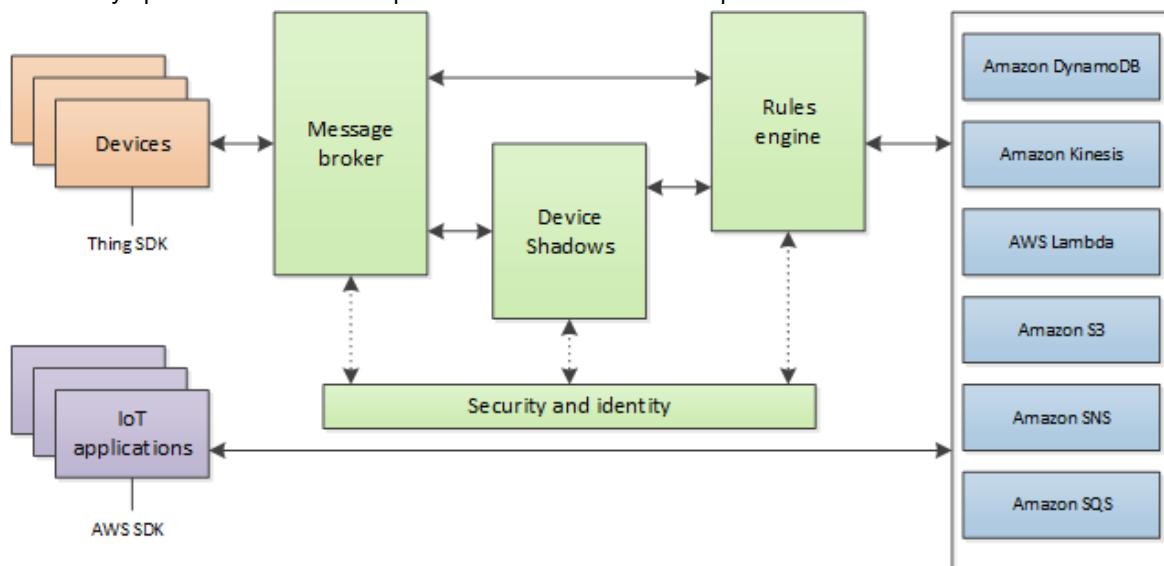
AWS IoT SiteWise recopila, almacena, organiza y supervisa los datos transmitidos desde equipos industriales mediante mensajes MQTT o API a escala proporcionando software que se ejecuta en una puerta de enlace de sus instalaciones. La puerta de enlace se conecta de forma segura a sus servidores de datos locales y automatiza el proceso de recopilación y organización de los datos y enviarlos a AWS Cloud.

AWS IoT Events

AWS IoT Events detecta y responde a los eventos de los sensores y aplicaciones de IoT. Los eventos son patrones de datos que identifican circunstancias más complicadas de lo esperado, como los detectores de movimiento que utilizan señales de movimiento para activar luces y cámaras de seguridad. AWS IoT Los eventos monitorizan continuamente los datos de múltiples sensores y aplicaciones de IoT y se integran con otros servicios, como AWS IoT Core, IoT SiteWise, DynamoDB y otros para permitir la detección temprana y conocimientos únicos.

Servicios de AWS IoT Core

AWS IoT Core proporciona los servicios que conectan sus dispositivos IoT a AWS Cloud para que otros servicios y aplicaciones en la nube puedan interactuar con los dispositivos conectados a Internet.



En la siguiente sección se describe cada uno de los AWS IoT Core servicios que se muestran en la ilustración.

AWS IoT Coreservicios de mensajería

La AWS IoT Core los servicios de conectividad proporcionan una comunicación segura con los dispositivos IoT y administran los mensajes que pasan entre ellos y AWS IoT.

Gateway de dispositivos

Permite a los dispositivos comunicarse de forma segura y eficaz con AWS IoT. La comunicación del dispositivo está protegida por protocolos seguros que utilizan certificados X.509.

Agente de mensajes

Proporciona un mecanismo seguro para que los dispositivos y las aplicaciones de AWS IoT publiquen y reciban mensajes entre sí. Puede usar el protocolo MQTT directamente o MQTT sobre WebSocket para publicar y suscribirse. Para obtener más información sobre los protocolos que AWS IoT soporta, consulte [the section called “Protocolos de comunicación de dispositivos” \(p. 81\)](#). Los dispositivos y los clientes también pueden usar la interfaz HTTP REST para publicar datos en el agente de mensajes.

El agente de mensajes distribuye los datos del dispositivo a los dispositivos que se han suscrito a él y a otros AWS IoT Coreservicios, como el servicio Device Shadow y el motor de reglas.

AWS IoT Core para LoRaWAN

AWS IoT Core para LoRaWAN permite configurar un privado LoRaWAN conectando su LoRa Dispositivos WAN y puertas de enlace para AWS sin necesidad de desarrollar y operar un LoRa Servidor de red WAN (LNS). Mensajes recibidos de LoRa Los dispositivos WAN se envían al motor de reglas, donde se pueden formatear y enviar a otros AWS IoT Servicios de .

Motor de reglas

El motor de reglas conecta los datos del agente de mensajes a otros AWS IoT servicios de almacenamiento y procesamiento adicional. Por ejemplo, puede insertar, actualizar o consultar una tabla de DynamoDB o invocar una función de Lambda basada en una expresión definida en el motor de reglas. Puede utilizar un lenguaje basado en SQL para seleccionar datos de cargas de mensajes y, a continuación, procesar y enviar datos a otros servicios, como Amazon Simple Storage Service (Amazon S3), Amazon DynamoDB y AWS Lambda. También puede crear reglas que vuelvan a publicar mensajes en el agente de mensajes y en otros suscriptores. Para obtener más información, consulte [Reglas para AWS IoT \(p. 472\)](#).

AWS IoT Coreservicios de control

La AWS IoT Core los servicios de control proporcionan funciones de seguridad, administración y registro de dispositivos.

Servicio de autenticación personalizado

Puede definir autorizadores personalizados para permitirle administrar su propia estrategia de autenticación y autorización con un servicio de autenticación personalizado y una función de Lambda. Los autorizadores personalizados permiten a AWS IoT autenticar sus dispositivos y autorizar operaciones mediante estrategias de autorización y autenticación de tokens al portador.

Los autorizadores personalizados pueden implementar varias estrategias de autenticación; por ejemplo, verificación de tokens web JSON o llamada de proveedor de OAuth. Deben devolver los documentos de la política que ha usado la puerta de enlace del dispositivo para autorizar las operaciones de MQTT. Para obtener más información, consulte [Autenticación personalizada \(p. 319\)](#).

Servicio de aprovisionamiento de dispositivos

Permite aprovisionar dispositivos mediante una plantilla que describe los recursos necesarios para el dispositivo: un objeto de cosa, un certificado y una o varias políticas. Un objeto objeto de

Un objeto es una entrada en el registro que contiene atributos que describen un dispositivo. Los dispositivos usan certificados para realizar la autenticación con AWS IoT. Las políticas determinan qué operaciones pueden realizar los dispositivos en AWS IoT.

Las plantillas contienen variables que se sustituirán por los valores en un diccionario (mapa). Puede usar la misma plantilla para aprovisionar varios dispositivos tan solo pasando diferentes valores para las variables de la plantilla en el diccionario. Para obtener más información, consulte [Aprovisionamiento de dispositivos \(p. 795\)](#).

Registro de grupos

Los grupos permiten administrar varios dispositivos a la vez clasificándolos en grupos. Los grupos también pueden contener otros grupos; puede crear una jerarquía de grupos. Cualquier acción que realice en un grupo principal se aplicará a sus grupos secundarios. La misma acción también se aplica a todos los dispositivos del grupo padre y a todos los dispositivos de los grupos secundarios. Los permisos concedidos a un grupo se aplicarán a todos los dispositivos del grupo y a todos sus grupos secundarios. Para obtener más información, consulte [Administración de dispositivos con AWS IoT \(p. 266\)](#).

Servicio de Jobs

Permite definir un conjunto de operaciones remotas que se envían a uno o más dispositivos conectados a o que se ejecutan en uno o más de esos dispositivos AWS IoT. Por ejemplo, puede definir un trabajo que indique a un conjunto de dispositivos descargar e instalar actualizaciones de firmware y aplicaciones, reiniciar, rotar certificados o realizar operaciones remotas de solución de problemas.

Para crear un trabajo, especifique una descripción de las operaciones remotas que se van a realizar y una lista de destinos que deben realizarlas. Los destinos pueden ser dispositivos individuales, grupos o ambos. Para obtener más información, consulte [Trabajos \(p. 675\)](#).

Registry (Registro)

Organiza los recursos asociados a cada dispositivo en la AWS Cloud. Es necesario registrar los dispositivos y asociar hasta tres atributos personalizados a cada uno. También es posible asociar certificados e ID de cliente MQTT a cada dispositivo para poder administrarlos y solucionar los problemas que presenten con mayor facilidad. Para obtener más información, consulte [Administración de dispositivos con AWS IoT \(p. 266\)](#).

Servicio de seguridad e identidad

Comparte la responsabilidad de la seguridad en la AWS Cloud. Los dispositivos deben proteger sus credenciales para enviar datos de forma segura al agente de mensajes. El agente de mensajes y el motor de reglas AWS funciones de seguridad para enviar datos de forma segura a dispositivos u otros AWS servicios. Para obtener más información, consulte [Autenticación \(p. 294\)](#).

Servicios de datos de AWS IoT Core

La AWS IoT Core los servicios de datos ayudan a sus soluciones de IoT a proporcionar una experiencia de aplicación fiable incluso con dispositivos que no siempre están conectados.

Sombra del dispositivo

Documento JSON utilizado para almacenar y recuperar información del estado actual de un dispositivo.

Servicio Device Shadow

El servicio Device Shadow mantiene el estado de un dispositivo para que las aplicaciones puedan comunicarse con un dispositivo, ya sea que el dispositivo esté conectado o no. Cuando un dispositivo

está sin conexión, el servicio Device Shadow administra sus datos para aplicaciones conectadas. Cuando el dispositivo se vuelve a conectar, sincroniza su estado con el de su sombra en el servicio Device Shadow. Los dispositivos también pueden publicar su estado actual en una sombra para que lo usen las aplicaciones o los demás dispositivos que podrían no estar conectados en todo momento. Para obtener más información, consulte [Servicio Device Shadow de AWS IoT \(p. 627\)](#).

AWS IoT Coreservicio de soporte técnico

Integración de Alexa Voice Service (AVS) paraAWS IoT

Lleva Alexa Voice a cualquier dispositivo conectado. AVS paraAWS IoT reduce el costo y la complejidad de la integración de Alexa. Esta característica utiliza AWS IoT para descargar tareas informáticas y de audio de memoria intensivas del dispositivo a la nube. Debido a la reducción resultante en el costo de la factura de materiales de ingeniería (EBOM), los fabricantes de dispositivos pueden incorporar Alexa en dispositivos IoT con recursos limitados de manera más económica para permitir a los consumidores hablar directamente con Alexa en su hogar, oficina o habitación del hotel y disfrutar así de una experiencia de ruido ambiente.

AVS paraAWS IoT permite la funcionalidad integrada de Alexa en MCU, como la clase ARM Cortex M con menos de 1 MB de RAM integrada. Para ello, AVS transfiere las tareas de memoria y computación a un dispositivo virtual de Alexa integrado en la nube. Esto reduce el costo de EBOM hasta en un 50%. Para obtener más información, consulte [Integración de Alexa Voice Service \(AVS\) paraAWS IoT \(p. 1257\)](#).

Integración de Amazon Sidewalk paraAWS IoT Core

Amazon Sidewalk es una red compartida que mejora las opciones de conectividad para ayudar a los dispositivos a funcionar mejor juntos. Amazon Sidewalk admite una amplia gama de dispositivos de clientes, como los que localizan mascotas o objetos de valor, aquellos que proporcionan seguridad doméstica inteligente y control de iluminación, y aquellos que proporcionan diagnósticos remotos para electrodomésticos y herramientas. Integración de Amazon Sidewalk paraAWS IoT Core permite que los fabricantes de dispositivos agreguen su flota de dispositivos Sidewalk alAWS IoT Cloud.

Para obtener más información, consulte [Integración de Amazon Sidewalk paraAWS IoT Core \(p. 1212\)](#)

Obtener más información sobre AWS IoT

Este tema te ayuda a familiarizarte con el mundo de AWS IoT. Puede obtener información general sobre cómo se aplican las soluciones de IoT en varios casos de uso, recursos de capacitación, enlaces a redes sociales paraAWS IoT y todos los demás AWS servicios y una lista de servicios y protocolos de comunicación que AWS IoT utiliza.

Recursos de formación paraAWS IoT

Ofrecemos estos cursos de formación para ayudarle a conocer mejor AWS IoT y cómo aplicarlos al diseño de su solución.

- [Introducción a AWS IoT](#)
Información general del vídeo sobre AWS IoT y sus servicios básicos.
- [Sumérgete en AWS IoT Autenticación y autorización](#)

Un curso avanzado que explora los conceptos de AWS IoT autenticación y autorización. Aprenderá a autenticar y autorizar a los clientes a acceder al AWS IoT API de plano de control y plano de datos.

- [Serie de la Fundación Internet de las cosas](#)

Una ruta de aprendizaje de los módulos de aprendizaje electrónico de IoT en diferentes tecnologías y funciones de IoT.

AWS IoTrecursos y guías

Se trata de recursos técnicos detallados sobre aspectos específicos de AWS IoT.

- [Objetivo de IoT —AWS IoTMarco de Well-Architected](#)

Documento que describe las prácticas recomendadas para diseñar aplicaciones IoT en AWS.

- [Diseño de temas MQTT para AWS IoT Core](#)

Un documento PDF que describe las prácticas recomendadas para diseñar temas MQTT en AWS IoT Corey apalancamiento AWS IoT Corefunciones con MQTT.

- [Fabricación y aprovisionamiento de dispositivos con certificados X.509 en AWS IoT Core](#)

Un documento PDF que describe las distintas formas en que AWS IoT prevé el aprovisionamiento de grandes flotas de dispositivos.

- [AWS IoT Core Device Advisor](#)

AWS IoT CoreDevice Advisor proporciona pruebas predefinidas que puede utilizar para validar dispositivos IoT para obtener prácticas recomendadas de conectividad fiables y seguras con AWS IoT Core, antes de desplegar dispositivos en producción.

- [Recursos de AWS IoT](#)

Recursos específicos de IoT, tales como guías técnicas, arquitecturas de referencia, libros electrónicos y publicaciones de blog seleccionadas, presentados en un índice que permite búsquedas.

- [Atlas de IoT](#)

Descripción general sobre cómo resolver problemas comunes de diseño de IoT. LaAtlas de IoTproporciona un análisis detallado de los desafíos de diseño que probablemente se enfrenta al desarrollar su solución de IoT.

- [AWSDocumentos técnicos y guías](#)

Nuestra colección actual de documentos técnicos y guías sobre AWS IoT y otros AWS tecnologías.

AWS IoT en redes sociales

Estos canales de redes sociales proporcionan información sobre AWS IoT y AWS-relacionados con.

- [El Internet de las cosas en AWS IoT— Blog oficial](#)

- [AWS IoTvídeos en el canal Amazon Web Services en YouTube](#)

Estas cuentas de redes sociales cubren todo AWS servicios, incluidos AWS IoT

- [El canal Amazon Web Services en YouTube](#)

- [Amazon Web Services en Twitter](#)

- [Amazon Web Services en Facebook](#)

- [Amazon Web Services en Instagram](#)

- [Amazon Web Services en LinkedIn](#)

AWSservicios utilizados por elAWS IoT Coremotor de reglas

LaAWS IoT Coreel motor de reglas se puede conectar a estosAWSServicios de .

- [Amazon DynamoDB](#)

Amazon DynamoDB es un servicio de bases de datos escalable y NoSQL que proporciona un rendimiento de bases de datos rápido y predecible.

- [Amazon Kinesis](#)

Amazon Kinesis facilita la recopilación, el procesamiento y el análisis de datos de streaming en tiempo real para que pueda obtener información oportuna y reaccionar rápidamente ante la nueva información. Amazon Kinesis puede ingerir datos en tiempo real como vídeo, audio, registros de aplicaciones, flujos de clics en sitios web y datos de telemetría de IoT para aprendizaje automático, análisis y otras aplicaciones.

- [AWS Lambda](#)

AWS Lambda le permite ejecutar código sin aprovisionar ni administrar servidores. Puede configurar el código de modo que se ejecute automáticamente enAWS IoTdatos y eventos o llámalo directamente desde una aplicación web o móvil.

- [Amazon Simple Storage Service](#)

Amazon Simple Storage Service (Amazon S3) puede almacenar y recuperar cualquier cantidad de datos en cualquier momento y desde cualquier parte de la web.AWS IoTLas reglas pueden enviar datos a Amazon S3 para almacenarlos.

- [Amazon Simple Notification Service](#)

Amazon Simple Notification Service (Amazon SNS) es un servicio web que permite a las aplicaciones, usuarios finales y dispositivos enviar y recibir notificaciones desde la nube.

- [Amazon Simple Queue Service](#)

Amazon Simple Queue Service (Amazon SQS) es un servicio de colas de mensajes que desacopla y escala microservicios, sistemas distribuidos y aplicaciones sin servidor.

- [AmazonOpenSearchService \(Servicio\)](#)

AmazonOpenSearchServicio (OpenSearchService) es un servicio administrado que facilita la implementación, el control y el escaladoOpenSearch, un popular motor de búsqueda y análisis de código abierto.

- [Amazon Machine Learning](#)

Amazon Machine Learning puede crear modelos de aprendizaje automático (ML) detectando pautas en los datos de IoT. El servicio utiliza estos modelos para procesar nuevos datos y generar predicciones para su aplicación.

- [Amazon CloudWatch](#)

AmazonCloudWatchproporciona una solución de monitorización fiable, escalable y flexible para ayudar a configurar, administrar y escalar sus propios sistemas de monitorización e infraestructura.

Protocolos de comunicación compatibles con AWS IoT Core

Estos temas proporcionan más información sobre los protocolos de comunicación utilizados por AWS IoT. Para obtener más información sobre los protocolos utilizados por AWS IoT para conectar dispositivos y servicios a AWS IoT, consulte [Conexión a AWS IoT Core \(p. 69\)](#).

- [MQTT \(Transporte de telemetría de Message Queue Server\)](#)

La página de inicio del sitio MQTT.org donde puede encontrar las especificaciones del protocolo MQTT. Para obtener más información sobre cómo AWS IoT admite MQTT, consulte [MQTT \(p. 85\)](#).

- [HTTPS \(protocolo de transferencia de hipertexto: seguro\)](#)

Los dispositivos y las aplicaciones pueden acceder a AWS IoT servicios mediante HTTPS.

- [LoRaWAN \(red de área extendida de largo alcance\)](#)

Los dispositivos WAN y las puertas de enlace se pueden conectar a AWS IoT utilizando AWS IoT Core para LoRaWAN.

- [TLS \(Transport Layer Security v1.2\)](#)

Especificación del TLS v1.2 (RFC 5246). AWS IoT utiliza TLS v1.2 para establecer conexiones seguras entre dispositivos y AWS IoT.

Novedades de AWS IoT consola

Estamos en proceso de actualización de la interfaz de usuario de AWS IoT consola para una nueva experiencia. Estamos actualizando la interfaz de usuario por etapas, por lo que algunas páginas de la consola tendrán una nueva experiencia, otras podrían tener la experiencia original y la nueva, y otras podrían tener solo la experiencia original.

En esta tabla se muestra el estado de las áreas individuales de la AWS IoT interfaz de usuario de consola desde el 27 de enero de 2022.

AWS IoT Estado de interfaz de usuario de consola

Página de la consola	Experiencia original	Nueva experiencia	Comentarios
Monitorización	No disponible	Disponible	
Actividad	No disponible	Disponible	
Incorporación-Introducción	No disponible	No se dispone todavía	
Incorporación: plantillas de aprovisionamiento de flotas	Disponible	No se dispone todavía	
Manejar- Objetos	Disponible	Disponible	
Manejar- Tipos	Disponible	Disponible	
Manejar- Grupos de objetos	Disponible	Disponible	

Página de la consola	Experiencia original	Nueva experiencia	Comentarios
Manejar- Grupos de facturación	Disponible	Disponible	
Manejar- Trabajos	Disponible	Disponible	
Manejar- Plantillas de Job	No disponible	Disponible	
Manejar- Tunelos	No disponible	Disponible	
Fleet Hub- Introducción	No disponible	Disponible	No se ofrece en todosRegiones de AWS
Fleet Hub- Aplicaciones	No disponible	Disponible	No se ofrece en todosRegiones de AWS
Greengrass- Introducción	No disponible	Disponible	No se ofrece en todosRegiones de AWS
Greengrass- Dispositivos de núcleo de	No disponible	Disponible	No se ofrece en todosRegiones de AWS
Greengrass- Componentes	No disponible	Disponible	No se ofrece en todosRegiones de AWS
Greengrass- Implementaciones	No disponible	Disponible	No se ofrece en todosRegiones de AWS
Greengrass- Clásico (V1)	Disponible	No disponible	No se ofrece en todosRegiones de AWS
Conectividad inalámbrica- Introducción	No disponible	Disponible	No se ofrece en todosRegiones de AWS
Conectividad inalámbrica- Gateways de	No disponible	Disponible	No se ofrece en todosRegiones de AWS
Conectividad inalámbrica- Dispositivos	No disponible	Disponible	No se ofrece en todosRegiones de AWS
Conectividad inalámbrica- Perfiles	No disponible	Disponible	No se ofrece en todosRegiones de AWS
Conectividad inalámbrica- Destinos	No disponible	Disponible	No se ofrece en todosRegiones de AWS
Seguridad- Certificados	Disponible	Disponible	
Seguridad- Políticas	Disponible	Disponible	
Seguridad- CA	Disponible	Disponible	
Seguridad- Alias de roles	Disponible	Disponible	

Página de la consola	Experiencia original	Nueva experiencia	Comentarios
Seguridad-Autorizadores	Disponible	Disponible	
Defender- Introducción	Disponible	No se dispone todavía	
Defender- Auditoría	Disponible	No se dispone todavía	
Defender- Detect	Disponible	No se dispone todavía	
Defender- Acciones de mitigación	Disponible	No se dispone todavía	
Defender- Configuración de	Disponible	No se dispone todavía	No se ofrece en todosRegiones de AWS
Acto- Reglas	Disponible	No se dispone todavía	
Acto- Destinos	Disponible	No se dispone todavía	
Pruebas- Device Advisor	Disponible	Disponible	No se ofrece en todosRegiones de AWS
Pruebas- Cliente de prueba MQTT	Disponible	Disponible	
Software	Disponible	No se dispone todavía	
Configuración	No disponible	Disponible	
Aprender	Disponible	No se dispone todavía	

Leyenda

Valores de estado

- Disponible

Se puede utilizar esta experiencia de interfaz de usuario.

- No disponible

Esta experiencia de interfaz de usuario no se puede utilizar.

- No se dispone todavía

Se está trabajando en la nueva experiencia de interfaz de usuario, pero aún no está lista.

- En curso

La nueva experiencia de interfaz de usuario está en proceso de actualizarse. Sin embargo, es posible que algunas páginas sigan teniendo la experiencia de usuario original.

Introducción a AWS IoT Core

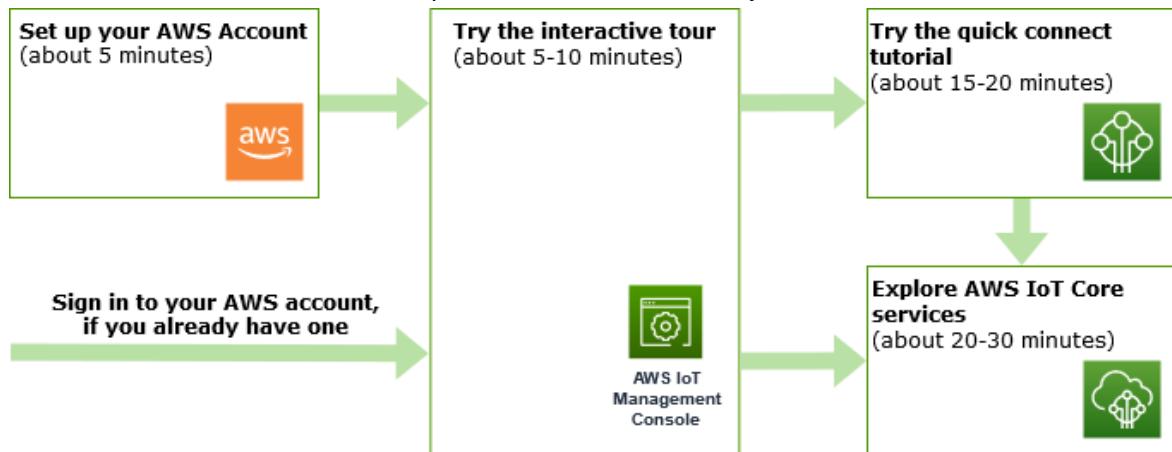
Tanto si eres nuevo en IoT como si tienes años de experiencia, estos recursos presentan elAWS IoTconceptos y términos que te ayudarán a empezar a usarAWS IoT.

- MirardentroAWS IoT sus componentes en[Cómo funciona AWS IoT \(p. 3\)](#).
- Aprendermás información sobreAWS IoT (p. 11)de nuestra colección de materiales de formación y videos. En este tema se incluye también una lista de servicios queAWS Iotpuede conectarse, enlaces a redes sociales y enlaces a especificaciones de protocolos de comunicación.
- [the section called “Connect del primer dispositivo aAWS IoT Core” \(p. 17\)](#).
- Desarrollarsus soluciones IoT por[Conexión a AWS IoT Core \(p. 69\)](#)y exploración de[Tutorial de AWS IoT \(p. 124\)](#).
- Probar y validarsus dispositivos IoT para una comunicación segura y fiable mediante el[Asesor de dispositivos \(p. 1058\)](#).
- Manejarsu solución medianteAWS IoT CoreServicios de administración, tales como[Indexación de flotas \(p. 825\)](#),[Trabajos \(p. 675\)](#), y[AWS IoT Device Defender \(p. 871\)](#).
- Analizarlos datos de tus dispositivos mediante el[Servicios de datos de AWS IoT \(p. 8\)](#).

Connect del primer dispositivo aAWS IoT Core

AWS IoT Coreservicios conectan dispositivos IoT aAWS IoTservicios y otrosAWSServicios de .AWS IoT Coreincluye la puerta de enlace del dispositivo y el agente de mensajes, que conectan y procesan mensajes entre los dispositivos IoT y la nube.

A continuación le mostramos cómo empezar a utilizarAWS IoT CoreyAWS IoT.



Esta sección presenta un recorrido por elAWS IoT Corepara presentar sus servicios clave y proporcionar varios ejemplos de cómo conectar un dispositivo aAWS IoT Corey pasar mensajes entre ellos. Pasar mensajes entre dispositivos y la nube es fundamental para todas las soluciones de IoT y es cómo sus dispositivos pueden interactuar con otrosAWSServicios de .

- [Configurar suCuenta de AWS \(p. 18\)](#)

Antes de poder usarAWS IoTservicios, debe configurar unCuenta de AWS. Si ya tiene unCuenta de AWSy un usuario de IAM para ti, puedes usarlos y omitir este paso.

- [Pruebe el tutorial interactivo \(p. 20\)](#)

Esta demostración es mejor si quieras ver qué básicoAWS IoTpuede funcionar sin conectar un dispositivo ni descargar ningún software. El tutorial interactivo presenta una solución simulada basada enAWS IoT Coreservicios que ilustran cómo interactúan.

- [Prueba el tutorial de conexión rápida \(p. 23\)](#)

Este tutorial es mejor si desea comenzar a utilizar rápidamenteAWS IoT vea cómo funciona en un escenario limitado. En este tutorial, necesitará un dispositivo e instalará algunosAWS IoTsoftware en él. Si no dispone de un dispositivo IoT, puede utilizar su computadora personal Windows, Linux o macOS como dispositivo para este tutorial. Si quieras probarAWS IoT, pero no dispone de un dispositivo, pruebe la siguiente opción.

- [ExplorarAWS IoT Coreservicios con un tutorial práctico \(p. 37\)](#)

Este tutorial es el mejor para los desarrolladores que desean empezar a utilizarAWS IoTpara que puedan seguir explorando otrosAWS IoT Corecaracterísticas como el motor de reglas y las sombras. Este tutorial sigue un proceso similar al tutorial de conexión rápida, pero proporciona más detalles sobre cada paso para permitir una transición más fluida a los tutoriales más avanzados.

- [Ver los mensajes MQTT con elAWS IoTCliente MQTT \(p. 65\)](#)

Aprenda a utilizar el cliente de prueba MQTT para ver cómo su primer dispositivo publica mensajes MQTT enAWS IoT. El cliente de pruebas MQTT es una herramienta útil para supervisar y solucionar problemas de conexiones de dispositivos.

Note

Si desea probar más de uno de estos tutoriales de introducción o repetir el mismo tutorial, debe eliminar el objeto de cosa que creó de un tutorial anterior antes de iniciar otro. Si no eliminas el objeto de cosa de un tutorial anterior, tendrás que usar otro nombre de cosa para tutoriales posteriores. Esto se debe a que el nombre de la cosa debe ser único en su cuenta yRegión de AWS.

Para obtener más información acerca de AWS IoT Core, consulte [¿Qué es AWS IoT Core? \(p. 1\)](#).

Configurar suCuenta de AWS

Antes de usar AWS IoT Core por primera vez, realice las siguientes tareas:

- [Registro en una Cuenta de AWS \(p. 18\)](#)
- [Cree un usuario y concédale permisos \(p. 19\)](#)
- [Abra el iconoAWS IoTconsola \(p. 20\)](#)

Si ya tiene unCuenta de AWSy un usuario de IAM para ti mismo, puedes usarlos y pasar a[the section called “Abra el iconoAWS IoTconsola” \(p. 20\).](#)

Registro en una Cuenta de AWS

Cuando te inscribes enAWS, su cuenta de se inscribe automáticamente en todos los servicios deAWS. Si tiene unCuenta de AWSya, omita este procedimiento. Si no dispone de una Cuenta de AWS, utilice el siguiente procedimiento para crear una.

Puede esperar pasar unos 5 minutos configurando su Cuenta de AWS.

Si no dispone de una Cuenta de AWS, siga los pasos que figuran a continuación para crear una.

Para registrarse en Cuenta de AWS

1. Abra <https://portal.aws.amazon.com/billing/signup>.
2. Siga las instrucciones en línea.

Parte del procedimiento de inscripción consiste en recibir una llamada telefónica e indicar un código de verificación en el teclado del teléfono.

Note

Guarde las cartasCuenta de AWSnúmero, porque lo necesitará en la siguiente tarea.

Cree un usuario y concédale permisos

Este procedimiento describe cómo crearse usted mismo un usuario de IAM y agregarlo a un grupo con permisos administrativos de una política administrada asociada. IAM es el iconoAWSservicio que administra los usuarios y el acceso aAWSde AWS. Debe hacerlo para que pueda crear elAWS IoTrecursos de tu cuenta y concédeles permiso para hacer lo que necesitan hacer.

Para crearse usted mismo un usuario administrador y agregarlo a un grupo de administradores (consola)

1. Inicie sesión en la [consola de IAM](#) como el propietario de la cuenta; para ello, elija Root user (Usuario raíz) e ingrese el email de su Cuenta de AWS. En la siguiente página, escriba su contraseña.

Note

Le recomendamos que siga la práctica recomendada de utilizar el usuario de IAM **Administrator** como se indica a continuación y guardar de forma segura las credenciales del usuario raíz. Inicie sesión como usuario raíz únicamente para realizar algunas [tareas de administración de servicios y de cuentas](#).

2. En el panel de navegación, elija Usuarios y, a continuación, elija Agregar usuarios.
3. En User name (Nombre de usuario), escriba **Administrator**.
4. Seleccione la casilla de verificación situada junto a AWS Management Console access (Acceso a la consola). A continuación, seleccione Custom password (Contraseña personalizada) y luego escriba la nueva contraseña en el cuadro de texto.
5. (Opcional) De forma predeterminada, AWS requiere al nuevo usuario que cree una nueva contraseña la primera vez que inicia sesión. Puede quitar la marca de selección de la casilla de verificación situada junto a User must create a new password at next sign-in (El usuario debe crear una nueva contraseña en el siguiente inicio de sesión) para permitir al nuevo usuario restablecer su contraseña después de iniciar sesión.
6. Seleccione Next (Siguiente): Permisos.
7. En Set permissions (Establecer permisos), elija Add user to group (Añadir usuario a grupo).
8. Elija Create group (Crear grupo).
9. En el cuadro de diálogo Create group (Crear grupo), en Group name (Nombre del grupo) escriba **Administrators**.
10. Elija Filter policies (Filtrar políticas) y, a continuación, seleccione AWS managed - job function (Función de trabajo administrada por AWS) para filtrar el contenido de la tabla.

11. En la lista de políticas, active la casilla de verificación AdministratorAccess. A continuación, elija Create group (Crear grupo).

Note

Debe activar el acceso de usuarios y roles de IAM a Facturación para poder utilizar los permisos AdministratorAccess para acceder a la consola de AWS Billing and Cost Management. Para ello, siga las instrucciones que se indican en el [paso 1 del tutorial sobre cómo delegar el acceso a la consola de facturación](#).

12. Retroceda a la lista de grupos y active la casilla de verificación del nuevo grupo. Elija Refresh si es necesario para ver el grupo en la lista.
13. Seleccione Next (Siguiente): Tags (Etiquetas).
14. (Opcional) Añadir metadatos al rol asociando las etiquetas como pares de clave-valor. Para obtener más información sobre el uso de etiquetas en IAM, consulte [Etiquetado de entidades de IAM](#) en la guía del usuario de IAM.
15. Seleccione Next (Siguiente): Review (Revisar)para ver la lista de suscripciones a grupos que se van a añadir al nuevo usuario. Cuando esté listo para continuar, elija Create user (Crear usuario).

Puede usar este mismo proceso para crear más grupos y usuarios, y para otorgar a los usuarios acceso a los recursos de la Cuenta de AWS. Para obtener información acerca de cómo usar las políticas que restringen los permisos de los usuarios a recursos de AWS específicos, consulte [Administración de accesos](#) y [Ejemplos de políticas](#).

Abra el iconoAWS IoTconsola

La mayoría de los temas orientados a las consolas de esta sección parten del[AWS IoTconsola](#). Si todavía no ha iniciado sesión en suCuenta de AWS, inicie sesión y, a continuación, abra el[AWS IoTconsola](#)y continúe con la siguiente sección para continuar conAWS IoT.

Pruebe el iconoAWS IoT Coretutorial interactivo

El tutorial interactivo muestra los componentes de una solución de IoT sencilla basada enAWS IoT. Las animaciones del tutorial muestran cómo interactúan los dispositivos IoT conAWS IoT CoreServicios de . En este tema se ofrece una vista previa delAWS IoT Coretutorial interactivo.

Para ejecutar la demostración, primero debe [the section called “Configurar suCuenta de AWS” \(p. 18\)](#). Sin embargo, el tutorial no requiere ningúnAWS IoTrecursos, software adicional o cualquier codificación.

Espere pasar unos 5-10 minutos en esta demostración. Darte 10 minutos te permitirá tener más tiempo para considerar cada uno de los pasos.

Para ejecutar laAWS IoT Coretutorial interactivo

1. Abra el icono[Centro de aprendizaje](#)en laAWS IoTconsola de .

En la páginaLe damos la bienvenida a laAWS IoTconsola, en laVea cómoAWS IoTfuncionalcon, elijalnicie el tutorial.

The screenshot shows the 'Welcome to the AWS IoT Console' page. At the top, there's a breadcrumb navigation: 'AWS IoT > Welcome to the AWS IoT Console'. Below the title, a message says 'To get started, you can jump into the recommended starting points below, or explore other learning resources as needed.' There are three main sections: 1) 'See how AWS IoT works' with four circular icons representing different IoT components, a 'Start the tutorial' button (labeled 'It takes 5 minutes'), and a note 'Explore an interactive tutorial through the components of AWS IoT.'. 2) 'Connect to AWS IoT' with a circular icon showing a network of nodes, a 'View connection options' button, and a note 'Connect a device, a mobile or web app to AWS IoT in a few easy steps!'. 3) 'Explore documentation' with an icon of a globe, a book, a laptop, and a smartphone, a 'Go to documentation' button, and a note 'The AWS IoT documentation is a great resource for more details.'

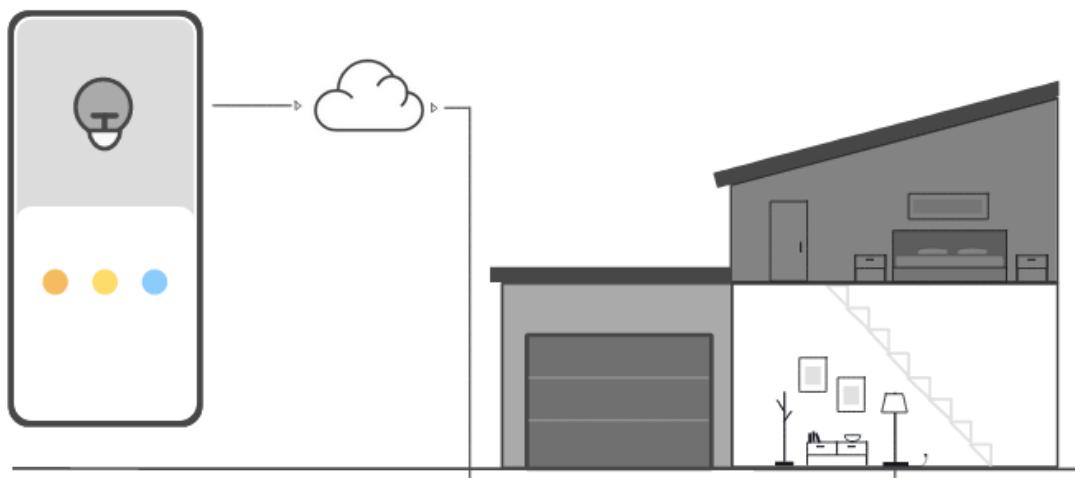
2. En el navegador AWS IoT Tutorial interactivo, revise las partes del tutorial y elija Explicación inicial cuando esté listo para continuar.

En las secciones siguientes, se describe cómo se AWS IoT Tutorial interactivo presenta estos AWS IoT Core Características de :

- Conexión de dispositivos IoT (p. 21)
- Guardar el estado del dispositivo sin conexión (p. 22)
- Enrutamiento de datos de dispositivos a servicios (p. 22)

Conexión de dispositivos IoT

Descubra cómo se comunican los dispositivos IoT con AWS IoT Core.



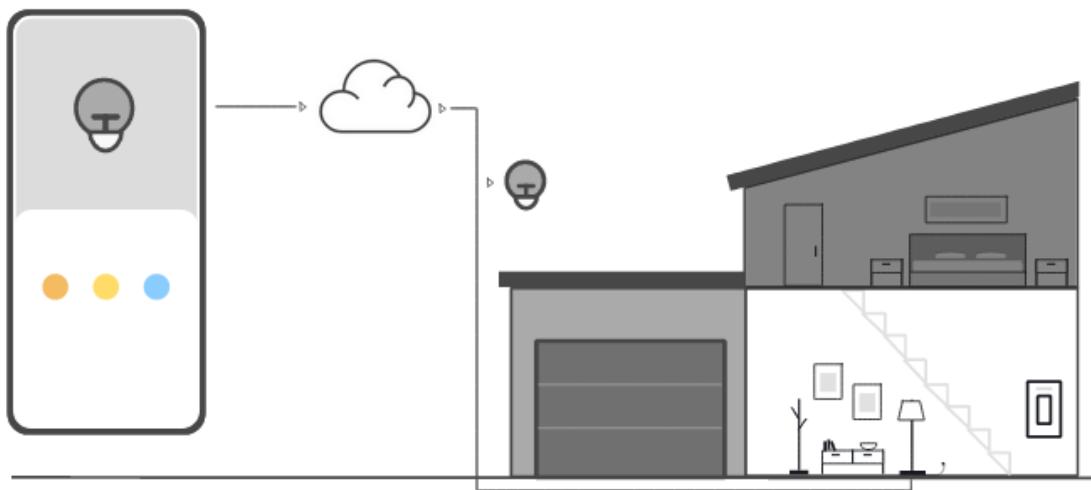
La animación de este paso muestra cómo dos dispositivos, el dispositivo de control de la izquierda y una lámpara inteligente en la casa a la derecha, se conectan y se comunican con AWS IoT Core en la nube. La animación muestra los dispositivos que se comunican con AWS IoT Core reaccionar a los mensajes que reciben.

La imagen de la consola incluye animaciones que no aparecen en esta imagen.

Para obtener más información acerca de la conexión de dispositivos a AWS IoT Core, consulte [Conexión a AWS IoT Core \(p. 69\)](#).

Guardar el estado del dispositivo sin conexión

Aprenda cómo AWS IoT Core guarda el estado del dispositivo mientras un dispositivo o una aplicación están desconectados.



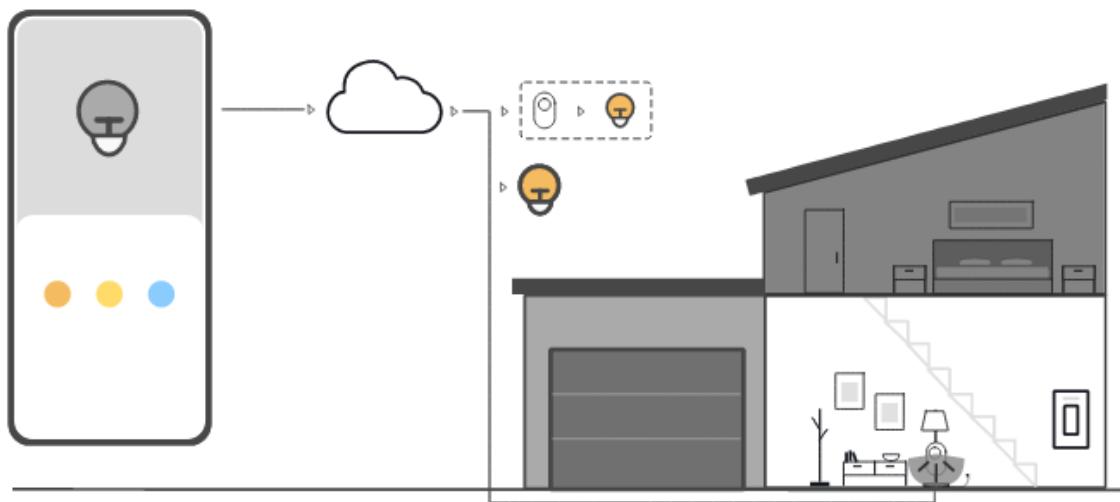
La animación de este paso muestra cómo el servicio Device Shadow en AWS IoT Core guarda la información del estado del dispositivo de control y la lámpara inteligente. Mientras la lámpara inteligente está desconectada, Device Shadow guarda los comandos del dispositivo de control.

Cuando la lámpara inteligente se vuelve a conectar a AWS IoT Core, recupera esos comandos. Cuando el dispositivo de control está desconectado, Device Shadow guarda la información de estado de la lámpara inteligente. Cuando el dispositivo de control se vuelve a conectar, recupera el estado actual de la lámpara inteligente para actualizar su pantalla.

Para obtener más información acerca de las sombras de dispositivos, consulte [Servicio Device Shadow de AWS IoT \(p. 627\)](#).

Enrutamiento de datos de dispositivos a servicios

Aprenda cómo AWS IoT Core envía el estado del dispositivo a otro AWS Servicio de .



La animación de este paso muestra cómo AWS IoT Core envía datos de los dispositivos a otros AWS servicios mediante AWS IoT reglas. AWS IoT las reglas se suscriben a mensajes específicos de los dispositivos, interpretan los datos de esos mensajes y enrutan los datos interpretados a otros servicios. En este ejemplo, un AWS IoT interpreta los datos de un sensor de movimiento y envía comandos a un Device Shadow, que luego los envía a la bombilla inteligente. Al igual que en el ejemplo anterior, Device Shadow almacena la información del estado del dispositivo de control.

Para obtener más información acerca de las reglas del AWS IoT, consulte [Reglas para AWS IoT \(p. 472\)](#).

Pruebe el iconoAWS IoTQuick Connect

En este tutorial, crearás tu primer objeto, conectarás un dispositivo a él y verás cómo envía mensajes MQTT.

Puede esperar dedicar entre 15 y 20 minutos a este tutorial.

Este tutorial es el mejor para las personas que desean empezar rápidamente AWS IoT para ver cómo funciona en un escenario limitado. Si estás buscando un ejemplo que te ayude a empezar para que puedas explorar más funciones y servicios, prueba [Explorar AWS IoT Coreservicios en tutorial práctico \(p. 37\)](#).

En este tutorial, descargarás y ejecutarás software en un dispositivo que se conecta a un recurso de objetos en AWS IoT Core como parte de una solución de IoT muy pequeña. El dispositivo puede ser un dispositivo IoT, como un Raspberry Pi, o también puede ser un equipo que ejecuta Linux, OS y OSX o Windows. Si desea conectar un dispositivo WAN de largo alcance (LoRaWAN) a AWS IoT, consulte el tutorial [Conexión de dispositivos y puertas de enlace a AWS IoT Core for LoRaWAN \(p. 1116\)](#).

Si el dispositivo admite un navegador que puede ejecutar la [AWS IoT consola](#), te recomendamos que completes este tutorial en ese dispositivo.

Note

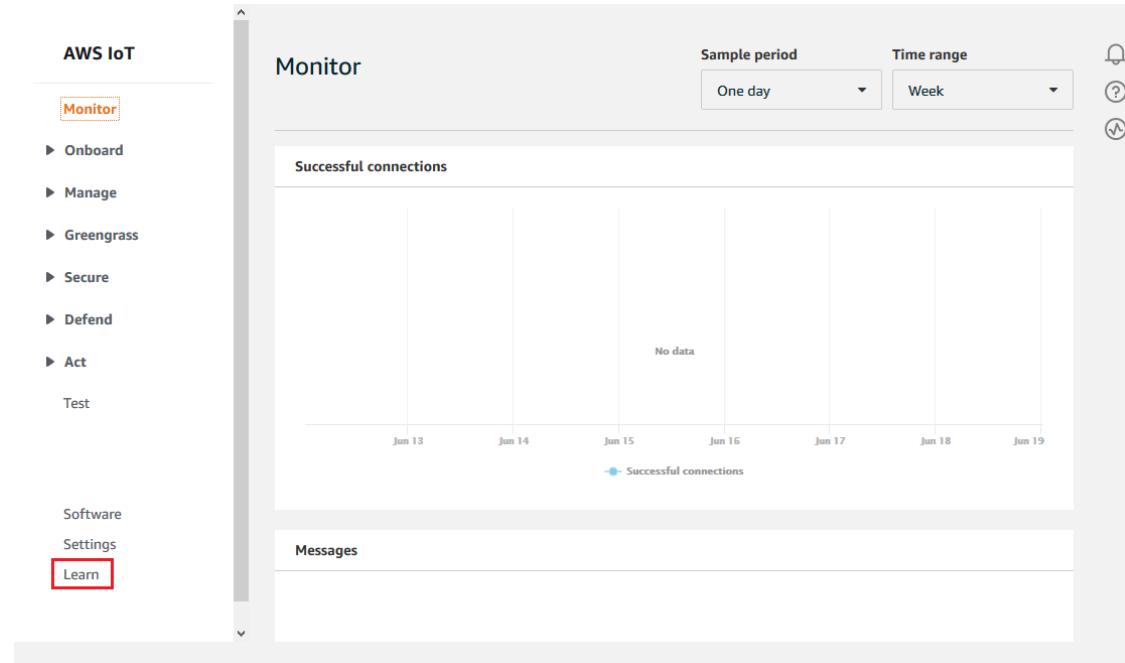
Si tu dispositivo no tiene un navegador compatible, sigue este tutorial en un ordenador. Cuando el procedimiento le pida que descargue el archivo, descárguelo en su equipo y, a continuación, transfiera el archivo descargado a su dispositivo mediante Secure Copy (SCP) o un proceso similar.

El tutorial requiere que su dispositivo IoT se comunique con el puerto 8443 de su cuenta de AWS endpoint de datos del dispositivo. Para comprobar si puede acceder a ese puerto, pruebe los procedimientos de [Comprobación de la conectividad con el punto final de datos de su dispositivo \(p. 33\)](#).

Paso 1. Inicie el tutorial

Si es posible, complete este procedimiento en su dispositivo; de lo contrario, esté preparado para transferir un archivo al dispositivo más adelante en este procedimiento.

1. Abra el icono [AWS IoT console](#) y en el menú de la izquierda, elija Aprender.



2. En la página [Connect to AWS IoT](#), elija Ver opciones de conexión.

The screenshot shows the AWS IoT Console homepage. On the left, a sidebar menu includes: Monitor, Onboard, Manage, Greengrass, Secure, Defend, Act, Test, Software, Settings, and Learn (which is highlighted). The main content area is titled "Welcome to the AWS IoT Console" with a sub-instruction: "To get started, you can jump into the recommended starting points below, or explore other learning resources as needed." It features three main sections: "See how AWS IoT works" (with icons for speech bubbles, a car, a shield, and a coin), "Connect to AWS IoT" (with an icon of a network graph), and "Explore documentation" (with icons of a book, a laptop, and a smartphone). Each section has a call-to-action button: "Start the tutorial", "View connection options", and "Go to documentation". A note at the bottom of the first section says "It takes 5 minutes".

3. En el navegadorIncorporación de un dispositivo con, elijaIntroducción.

This screenshot shows the "Connect to AWS IoT" section of the AWS IoT Console. The sidebar remains the same. The main content is titled "Connect to AWS IoT". It contains two main sections: "Onboard a device" (illustrated with icons of a car, a windmill, and a thermometer) and "Onboard many devices" (illustrated with a box containing a gear). Each section has a call-to-action button: "Get started" (highlighted with a red box) and "Create template". A note in the "Onboard a device" section says "It takes 5 minutes".

4. Revise los pasos que describen lo que harás en este tutorial. Cuando esté listo para continuar, elijaIntroducción.

Connect to AWS IoT

Connecting a device (like a development kit or your computer) to AWS IoT requires the completion of the following steps. In this process you will:

-  **Register a device**
A thing is the representation and record of your physical device in the cloud. Any physical device needs a thing record in order to work with AWS IoT.
-  **Download a connection kit**
The connection kit includes some important components: **security credentials, the SDK of your choice, and a sample project**.
-  **Configure and test your device**
Using the connection kit, you will configure your device by **transferring files and running a script**, and **test that it is connected** to AWS IoT correctly.

Want to learn more about the components of AWS IoT?
[Try the interactive overview](#)

[Get started](#)

Paso 2. Crear objeto objeto objeto

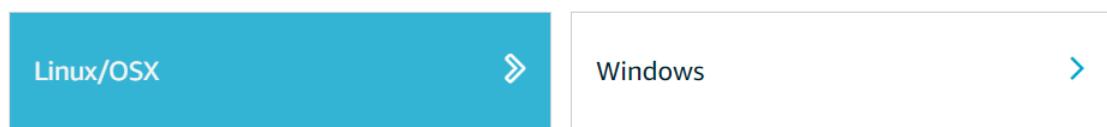
1. En la página ¿Cómo te conectas a AWS IoT?, elija la plataforma y el idioma de la AWS IoTSDK de dispositivo que desea utilizar. En este ejemplo se utiliza la plataforma Linux/OSX y el SDK de Python. Asegúrese de que tiene instalado python3 y pip3 en su dispositivo de destino antes de continuar con el siguiente paso.

Note

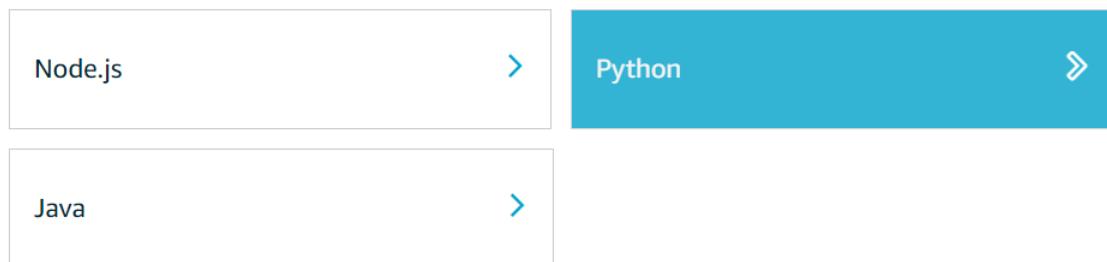
Asegúrese de consultar la lista de software de requisitos previos que necesita el SDK elegido en la parte inferior de la página de la consola.

Debe tener instalado el software necesario en el equipo de destino antes de continuar con el siguiente paso.

Después de elegir el idioma del SDK de la plataforma y del dispositivo, elija Próximo.



Choose a AWS IoT Device SDK



Some prerequisites to consider:

the device should have **Python** and **Git** installed and a **TCP connection to the public internet on port 8883**.

Looking for AWS IoT Device SDKs and documentation?
[View AWS IoT Device SDKs](#)

Next

2. En el navegadorNombre, escriba el nombre de su objeto de cosa. El nombre de la cosa que se utiliza en este ejemplo es**MyIoTThing**.

Important

Vuelve a comprobar el nombre de tu cosa antes de continuar.

El nombre de una cosa no se puede cambiar después de crear el objeto de cosa. Si desea cambiar el nombre de una cosa, debe crear un objeto objeto de cosa nuevo con el nombre correcto de cosa y eliminar después el que tenga el nombre incorrecto.



A thing is the representation and record of your physical device in the cloud. Any physical device needs a thing to work with AWS IoT. Creating a thing will also create a thing shadow.

[Choose an existing thing instead?](#)

Name

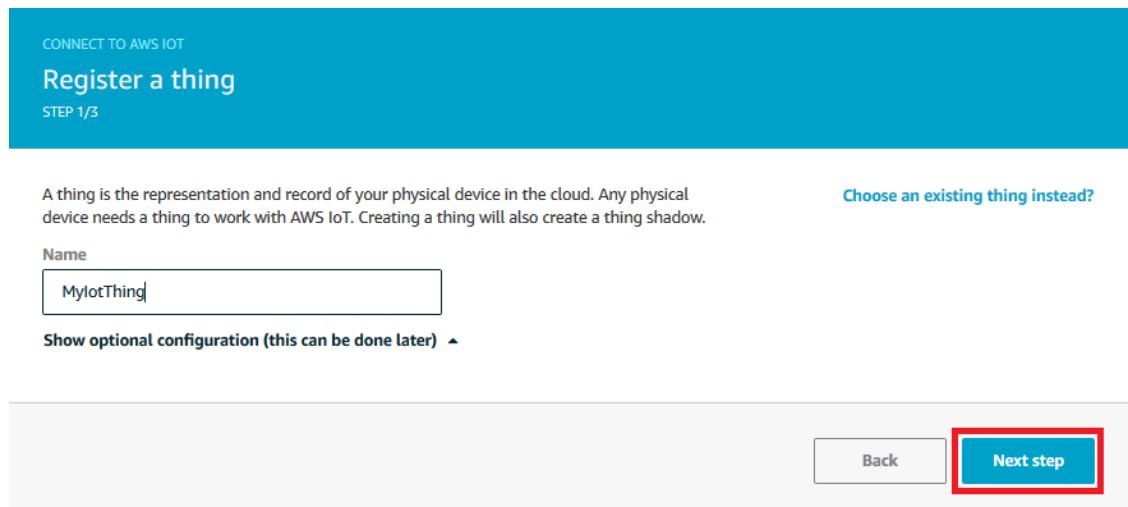
Give your thing a name

Show optional configuration (this can be done later) ▾

Back

Next step

3. Despues de darle un nombre a su objeto de cosa, elijaPaso siguiente.



Paso 3. Descarga archivos en tu dispositivo

Esta página aparece después de que AWS IoT ha creado el kit de conexión, que incluye los siguientes archivos y recursos que necesita su dispositivo:

- Los archivos de certificado de la cosa que se utilizan para autenticar el dispositivo
 - Un recurso de política para autorizar a su objeto de cosa a interactuar con AWS IoT
 - El guión para descargar el AWSSDK del dispositivo y ejecute el programa de ejemplo en su dispositivo
1. Cuando esté listo para continuar, elija la opción 'Descargar el kit de conexión para' para descargar el kit de conexión para la plataforma que eligió anteriormente.

CONNECT TO AWS IOT

Download a connection kit

STEP 2/3

The following AWS IoT resources will be created:

A thing in the AWS IoT registry	MyiotThing	
A policy to send and receive messages	MyiotThing-Policy	Preview policy

The connection kit contains:

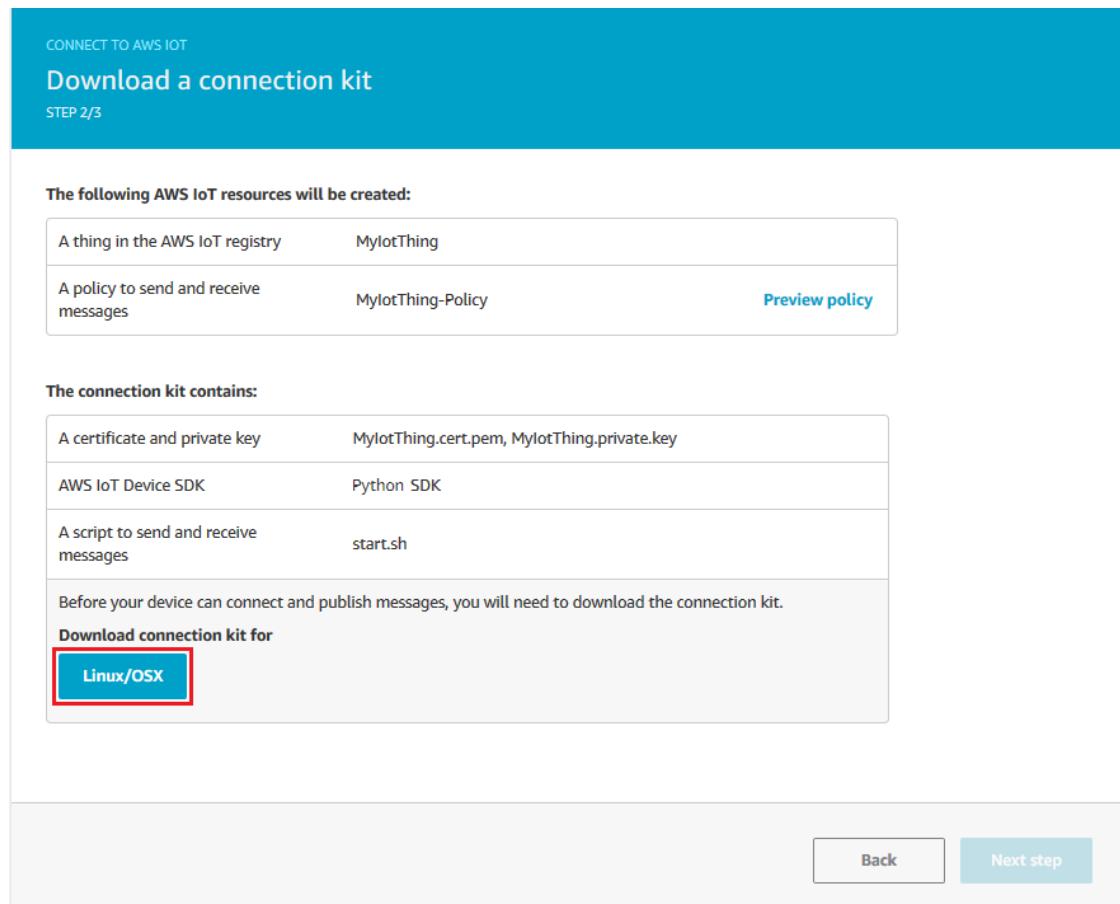
A certificate and private key	MyiotThing.cert.pem, MyiotThing.private.key
AWS IoT Device SDK	Python SDK
A script to send and receive messages	start.sh

Before your device can connect and publish messages, you will need to download the connection kit.

Download connection kit for

[Linux/OSX](#)

[Back](#) [Next step](#)



2. Si está ejecutando este procedimiento en su dispositivo, guarde el archivo del kit de conexión en un directorio desde el que puede ejecutar comandos de línea de comandos.
Si no está ejecutando este procedimiento en su dispositivo, guarde el archivo del kit de conexión en un directorio local y, a continuación, transfiera el archivo al dispositivo.
3. Despues de tener el archivo del kit de conexión en el dispositivo, continúe con el tutorial seleccionando Paso siguiente.

CONNECT TO AWS IOT

Download a connection kit

STEP 2/3

The following AWS IoT resources will be created:

A thing in the AWS IoT registry	MyiotThing	
A policy to send and receive messages	MyiotThing-Policy	Preview policy

The connection kit contains:

A certificate and private key	MyiotThing.cert.pem, MyiotThing.private.key
AWS IoT Device SDK	Python SDK
A script to send and receive messages	start.sh

Before your device can connect and publish messages, you will need to download the connection kit.

Download connection kit for

[Linux/OSX](#)

[Back](#) Next step

Paso 4: Ejecución de la muestra

Este procedimiento se realiza en un terminal o ventana de comandos del dispositivo mientras sigue las instrucciones que se muestran en la consola. Los comandos que ve en la consola son para el sistema operativo que eligió en the section called “Paso 2. Crear objeto objeto” (p. 26). Los que se muestran aquí son para los sistemas operativos Linux/OSX.

1. En un terminal o ventana de comandos del dispositivo, en el directorio con el archivo del kit de conexión, lleve a cabo los pasos que se muestran en el AWS IoTconsola de .

Si utiliza un Windows PowerShell ventana de comandos yunzipcomando no funciona, reemplazaunzipconexpand-archivee intente de nuevo la línea de comandos.

To configure and test the device, perform the following steps.

Step 1: Unzip the connection kit on the device

```
unzip connect_device_package.zip
```

Step 2: Add execution permissions

```
chmod +x start.sh
```

Step 3: Run the start script. Messages from your thing will appear below

```
./start.sh
```

Waiting for messages from your device

Back Done

- Después de introducir el comando desde Paso 3 en la consola, debería ver una salida en el terminal o en la ventana de comandos del dispositivo similar a la siguiente. Esta salida proviene de los mensajes a los que el programa envía y luego recibe de ellos.AWS IoT Core.

```
Received a new message:  
{"message": "Hello World!", "sequence": 1}  
from topic:  
sdk/test/Python  
-----  
  
2021-12-18 00:19:00,752 - AWSShadowPythonSDK.core.protocol.internal.clients - DEBUG - Invoking custom event callback...  
2021-12-18 00:19:01,757 - AWSShadowPythonSDK.core.protocol.mqtt_core - INFO - Performing sync publish...  
2021-12-18 00:19:01,757 - AWSShadowPythonSDK.core.protocol.internal.clients - DEBUG - Filling in custom puback (QoS>0) event callback...  
2021-12-18 00:19:01,841 - AWSShadowPythonSDK.core.protocol.internal.workers - DEBUG - Produced [puback] event  
2021-12-18 00:19:01,847 - AWSShadowPythonSDK.core.protocol.internal.workers - DEBUG - Dispatching [puback] event  
2021-12-18 00:19:01,847 - AWSShadowPythonSDK.core.protocol.internal.clients - DEBUG - Invoking custom event callback...  
2021-12-18 00:19:01,847 - AWSShadowPythonSDK.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 - AWSShadowPythonSDK.core.protocol.internal.workers - DEBUG - Produced [message] event  
2021-12-18 00:19:01,875 - AWSShadowPythonSDK.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 - AWSShadowPythonSDK.core.protocol.internal.clients - DEBUG - Invoking custom event callback...  
2021-12-18 00:19:02,871 - AWSShadowPythonSDK.core.protocol.mqtt_core - INFO - Performing sync publish...  
2021-12-18 00:19:02,871 - AWSShadowPythonSDK.core.protocol.internal.clients - DEBUG - Filling in custom puback (QoS>0) event callback...  
2021-12-18 00:19:02,956 - AWSShadowPythonSDK.core.protocol.internal.workers - DEBUG - Produced [puback] event  
2021-12-18 00:19:02,960 - AWSShadowPythonSDK.core.protocol.internal.workers - DEBUG - Dispatching [puback] event  
2021-12-18 00:19:02,960 - AWSShadowPythonSDK.core.protocol.internal.clients - DEBUG - Invoking custom event callback...  
2021-12-18 00:19:02,960 - AWSShadowPythonSDK.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 - AWSShadowPythonSDK.core.protocol.internal.workers - DEBUG - Produced [message] event  
2021-12-18 00:19:02,987 - AWSShadowPythonSDK.core.protocol.internal.workers - DEBUG - Dispatching [message] event  
Received a new message:  
{"message": "Hello World!", "sequence": 3}  
from topic:  
sdk/test/Python
```

Mientras se ejecuta el programa de ejemplo, en Paso 4: Enviar un mensaje al dispositivo, introduzca un mensaje comoHello World!en laAWS IoTconsola de . Para enviar el mensaje, elijaMándame. El mensaje de prueba aparece en el terminal o en la ventana de comandos del dispositivo.

Note

Para obtener más información sobre la suscripción y publicación de temas, consulte el código de ejemplo del SDK elegido.

- Para ejecutar de nuevo el programa de ejemplo, puede repetir los comandos desde Paso 3/3en la consola de este procedimiento.

4. (Opcional) Si desea ver los mensajes de su cliente de IoT en laAWS IoTconsola, abra el icono[Cliente de pruebas MQTT](#)en elPruebasPágina de laAWS IoTconsola de . Si eligió el SDK de Python, en elCliente de pruebas MQTT, enFiltro de temas, introduzca el tema, como[sdlk/test/python](#)para suscribirse a los mensajes desde el dispositivo. Los filtros de temas distinguen entre mayúsculas y minúsculas y dependen del lenguaje de programación del SDK que elijas enPaso 1/1. Para obtener más información sobre la suscripción y publicación de temas, consulte el ejemplo de código del SDK elegido.
5. Después de suscribirse al tema de prueba, ejecute./start.shen el dispositivo. Para obtener más información, consulte [the section called “Ver los mensajes MQTT con el cliente MQTT de AWS IoT” \(p. 65\).](#)

Después de correr./start.sh, aparecen mensajes en el cliente MQTT de, similar a la siguiente:

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

Lasequenceincrementos numéricos en uno cada vez que un nuevoHello Worldse recibe el mensaje y se detiene cuando finaliza el programa.

6. Para finalizar el tutorial y ver un resumen, en elAWS IoTConsola de, elijaTerminado.

Connected successfully

A device was connected to AWS IoT by performing some tasks in AWS IoT and on the device.



Registered a thing to represent a device in AWS IoT

[Learn more](#)



Set up security for the device using a certificate and policy

[Learn more](#)



Used a device SDK to connect a device to AWS IoT

[Learn more](#)



Received messages from the device

[Learn more](#)

Want to learn more about the components of AWS IoT?
[Try the interactive overview](#)

[Done](#)

Paso 5. Exploración más

Estas son algunas ideas para explorarAWS Iotdespués de completar el inicio rápido.

- [Ver mensajes MQTT en el cliente MQTT](#)

Desde el icono [AWS IoT consola](#), puede abrir [Cliente MQTT en el Prueba](#) en la Página de la AWS IoT consola de . En el navegador [Cliente MQTT](#), suscribirse a #y, a continuación, en su dispositivo, ejecute el programa ./start.sh como se describe en el paso anterior. Para obtener más información, consulte [the section called “Ver los mensajes MQTT con el cliente MQTT de AWS IoT” \(p. 65\)](#).

- Ejecuta pruebas en tus dispositivos con [Device Advisor](#)

Utilice Device Advisor para comprobar si sus dispositivos pueden conectarse de forma segura y fiable e interactuar con ellos, AWS IoT.

- [the section called “Pruebe el icono AWS IoT Core tutorial interactivo” \(p. 20\)](#)

Para iniciar el tutorial interactivo, desde el [Aprender](#) en la Página de la AWS IoT consola de, en el directorio [Vea](#) cómo AWS IoT funcional con, elija [Inicie el tutorial](#).

- [Prepárate para explorar más tutoriales \(p. 37\)](#)

Este inicio rápido te ofrece solo una muestra de AWS IoT. Si quieras explorar AWS IoT más información sobre las características que la convierten en una potente plataforma de soluciones IoT, comience a preparar su plataforma de desarrollo mediante [Explorar AWS IoT Cores servicios en tutorial práctico \(p. 37\)](#).

Comprobación de la conectividad con el punto final de datos de su dispositivo

En este tema se describe cómo probar la conexión de un dispositivo con la endpoint de datos de dispositivo, el punto final al que utilizan los dispositivos IoT para conectarse AWS IoT.

Realice estos procedimientos en el dispositivo que desea probar o mediante una sesión de terminal SSH conectada al dispositivo que desea probar.

Para probar la conectividad de un dispositivo con el punto final de datos del dispositivo.

- [Buscar el punto final de datos de su dispositivo \(p. 33\)](#)
- [Comprobación de la conexión rápidamente \(p. 34\)](#)
- [Haga que la aplicación pruebe la conexión con el puerto y el extremo de datos de su dispositivo \(p. 34\)](#)
- [Probar la conexión con el puerto y el extremo de datos del dispositivo \(p. 36\)](#)

Buscar el punto final de datos de su dispositivo

Para encontrar el punto de enlace de datos de su dispositivo

1. En el navegador [AWS IoT consola](#), cerca de la parte inferior del panel de navegación, elija [Configuración](#).
2. En el navegador [Configuración](#), en la [Punto final de datos de dispositivo](#) contenedor, localice el [Punto de enlace](#) valorar y copiarlo.
3. El valor de punto de enlace es exclusivo de su cuenta de AWS. Es similar a este ejemplo: a3qEXAMPLEsffp-ats.iot.eu-west-1.amazonaws.com.

Guarde el punto final de datos del dispositivo para utilizarlo en los siguientes procedimientos.

Comprobación de la conexión rápidamente

Este procedimiento comprueba la conectividad general con el extremo de datos del dispositivo, pero no prueba el puerto específico que utilizarán los dispositivos. Esta prueba utiliza un programa común y, por lo general, es suficiente para saber si los dispositivos se pueden conectar aAWS IoT.

Si desea probar la conectividad con el puerto específico que utilizarán sus dispositivos, omita este procedimiento y continúe[Haga que la aplicación pruebe la conexión con el puerto y el extremo de datos de su dispositivo \(p. 34\)](#).

Para probar rápidamente el punto final de datos del dispositivo

1. En una ventana de terminal o línea de comandos del dispositivo, sustituya el punto final de datos del dispositivo de ejemplo ([a3qEXAMPLEsffp-ats.iot.eu-west-1.amazonaws.com](#)) con el punto final de datos del dispositivo de su cuenta y, a continuación, introduzca este comando.

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

2. Si ping muestra una salida similar a la siguiente, que se ha conectado correctamente al punto final de datos del dispositivo. Si bien no se comunicaba conAWS IoT directamente, encontró el servidor y es probable queAWS IoT esté disponible a través de este endpoint.

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

Si está satisfecho con este resultado, puede dejar de probar aquí.

Si desea probar la conectividad con el puerto específico utilizado porAWS IoT, siga en[Haga que la aplicación pruebe la conexión con el puerto y el extremo de datos de su dispositivo \(p. 34\)](#).

3. Si ping devolvió una salida correcta, compruebe el valor del endpoint para asegurarse de que tiene el endpoint correcto y compruebe la conexión del dispositivo a Internet.

Haga que la aplicación pruebe la conexión con el puerto y el extremo de datos de su dispositivo

Se puede realizar una prueba de conectividad más exhaustiva mediante nmap. Este procedimiento comprueba si nmap se instaló en el dispositivo.

Para comprobar si hay nmap en el dispositivo

1. En una ventana de terminal o línea de comandos del dispositivo que desea probar, introduzca este comando para ver si nmap está instalado.

```
nmap --version
```

2. Si aparece una salida similar a la siguiente, nmap está instalado y puede continuar en[the section called "Probar la conexión con el puerto y el extremo de datos del dispositivo" \(p. 36\)](#).

```
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. Si no ve una respuesta similar a la que se muestra en el paso anterior, debe instalar nmap en el dispositivo. Seleccione el procedimiento del sistema operativo del dispositivo.

Linux

Este procedimiento requiere tener permiso para instalar software en el equipo.

Para instalar nmap en su equipo Linux

1. En una ventana de terminal o línea de comandos del dispositivo, introduzca el comando que corresponde a la versión de Linux que está ejecutando.

- a. Debian o Ubuntu:

```
sudo apt install nmap
```

- b. CentOS o RHEL:

```
sudo yum install nmap
```

2. Pruebe la instalación con este comando:

```
nmap --version
```

3. Si aparece una salida similar a la siguiente, nmap está instalado y puede continuar en [the section called “Probar la conexión con el puerto y el extremo de datos del dispositivo” \(p. 36\)](#).

```
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

Este procedimiento requiere tener permiso para instalar software en el equipo.

Para instalar nmap en su equipo MacOS

1. En un navegador, abra <https://nmap.org/download#macosx> y descargue el último estable instalador. Cuando se le pregunte, seleccione Abrir con DiskImageInstaller.
2. En la ventana de instalación, mueva el paquete a la Aplicaciones folder.
3. En el navegador Buscador, localice lanmap-xxxx-mpkgen la Aplicaciones folder. Ctrl-click el paquete en y seleccione Abiertopara abrir el paquete.
4. Revise el cuadro de diálogo de seguridad. Si está preparado para instalar nmap, elige Abierto para instalar nmap.

5. EnTerminal, pruebe la instalación con este comando.

```
nmap --version
```

6. Si aparece una salida similar a la siguiente,nmapestá instalado y puede continuar enthe section called “Probar la conexión con el puerto y el extremo de datos del dispositivo” (p. 36).

```
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 nssock engines: kqueue poll select
```

Windows

Este procedimiento requiere tener permiso para instalar software en el equipo.

Para instalar nmap en su equipo Windows

1. En un navegador, abra<https://nmap.org/download#windows> descargueEl último establolanzamiento del programa de configuración.

Si se le solicita, elijaGuardar archivo de. Una vez descargado el archivo, ábrelo desde la carpeta de descargas.

2. Cuando finalice la descarga del archivo de configuración, abra descargadonmap-xxxx-setup.exePara instalar la aplicación.
3. Acepte la configuración predeterminada a medida que se instala el programa.

No necesita la aplicación Npcap para esta prueba. Puede anular la selección de esta opción si no desea instalarla.

4. EnCommand, pruebe la instalación con este comando.

```
nmap --version
```

5. Si aparece una salida similar a la siguiente,nmapestá instalado y puede continuar enthe section called “Probar la conexión con el puerto y el extremo de datos del dispositivo” (p. 36).

```
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 nssock engines: iocp poll select
```

Probar la conexión con el puerto y el extremo de datos del dispositivo

Para probar el puerto y el extremo de datos del dispositivo

1. En una ventana de terminal o línea de comandos del dispositivo, sustituya el punto final de datos del dispositivo de ejemplo (<a3qEXAMPLESffp-ats.iot.eu-west-1.amazonaws.com>) con el punto final de datos del dispositivo de su cuenta y, a continuación, introduzca este comando.

```
nmap -p 8443 a3qEXAMPLESffp-ats.iot.eu-west-1.amazonaws.com
```

2. Si nmap muestra una salida similar a la siguiente, nmap se ha podido conectar correctamente al extremo de datos del dispositivo en el puerto seleccionado.

```
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
```

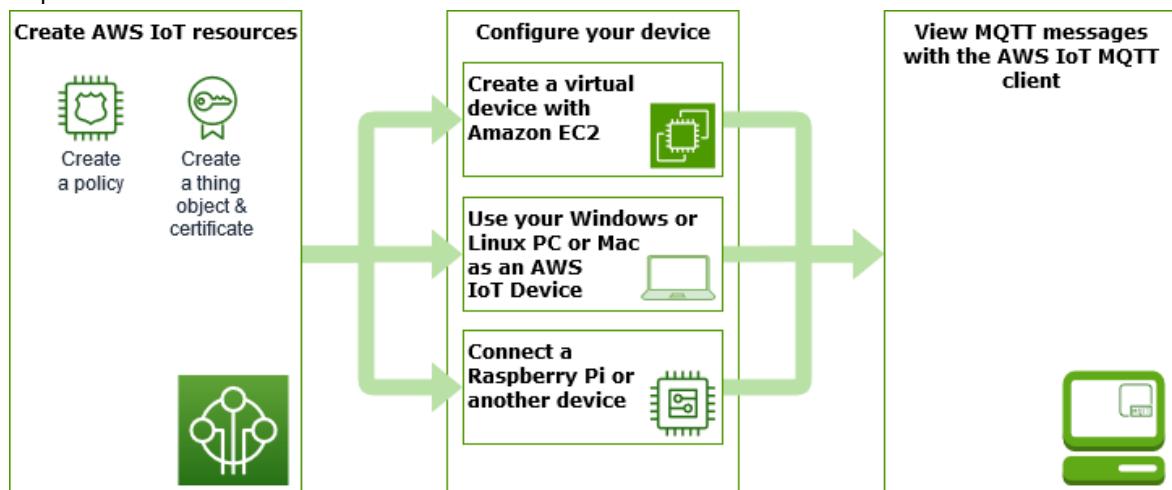
3. Si nmap ha devuelto una salida correcta, compruebe el valor del endpoint para asegurarse de que tiene el endpoint correcto y compruebe la conexión de su dispositivo a Internet.

Puede probar otros puertos en el extremo de datos del dispositivo, como el puerto 443, el puerto HTTPS principal, sustituyendo el puerto utilizado en el paso 1, **8443**, con el puerto que desea probar.

ExplorarAWS IoT Coreservicios en tutorial práctico

En este tutorial, instalará el software y creará los recursos necesarios para conectar un dispositivo a AWS IoT para que pueda enviar y recibir mensajes MQTT con AWS IoT Core. Verá los mensajes en el cliente MQTT en la AWS IoT consola de .

Puede esperar pasar de 20 a 30 minutos en este tutorial. Si utiliza un dispositivo IoT o una Raspberry Pi, este tutorial puede tardar más si, por ejemplo, necesita instalar el sistema operativo y configurar el dispositivo.



Este tutorial es el mejor para los desarrolladores que desean empezar a utilizar AWS IoT para que puedan seguir explorando funciones más avanzadas, como el motor de reglas y las sombras. Este tutorial te prepara para seguir aprendiendo sobre AWS IoT Core y cómo interactúa con otros AWS explicando los pasos con más detalle que el tutorial de inicio rápido. Si está buscando un poco rápido, Hello World, experimenta, prueba el [Pruebe el icono AWS IoT Quick Connect \(p. 23\)](#).

Después de configurar su cuenta de AWS IoT, seguirás estos pasos para ver cómo conectar un dispositivo y hacer que envíe mensajes a AWS IoT.

Pasos siguientes

- [Elige qué opción de dispositivo es la mejor para ti \(p. 38\)](#)
- [the section called “CrearAWS IoTRecursos” \(p. 38\) si no va a crear un dispositivo virtual con Amazon EC2.](#)
- [the section called “Configuración del dispositivo” \(p. 42\)](#)
- [the section called “Ver los mensajes MQTT con el cliente MQTT de AWS IoT” \(p. 65\)](#)

Para obtener más información acerca de AWS IoT Core, consulte [¿Qué es AWS IoT Core? \(p. 1\)](#).

¿Qué opción de dispositivo es la mejor para ti?

Si no estás seguro de qué opción elegir, utiliza la siguiente lista de las ventajas y desventajas de cada opción para ayudarte a decidir cuál es la mejor para ti.

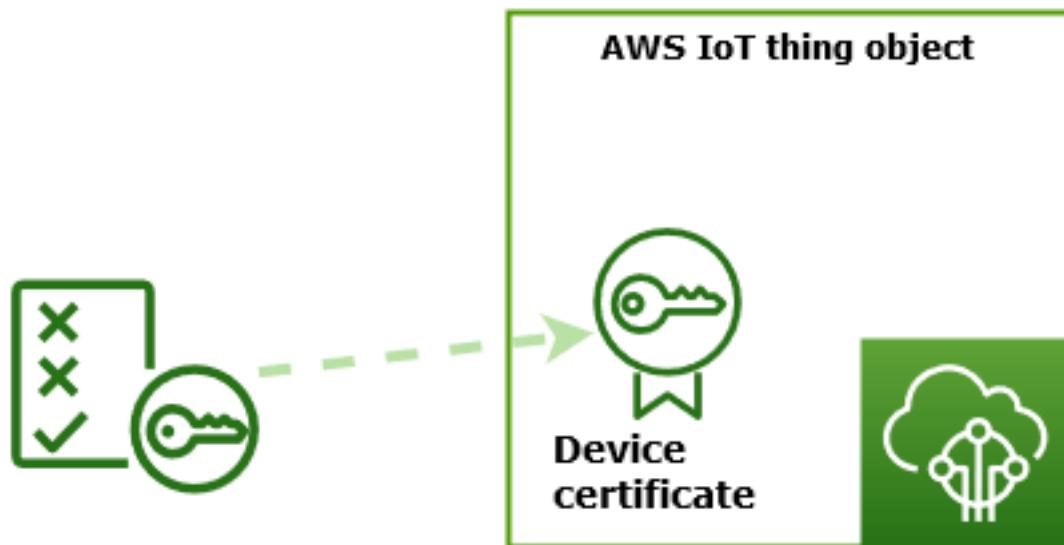
Opción	Esta podría ser una buena opción si:	Puede que esta no sea una buena opción si:
the section called “Creación de un dispositivo virtual con Amazon EC2” (p. 42)	<ul style="list-style-type: none">• No tienes tu propio dispositivo para probar.• No desea instalar software en su propio sistema.• Desea realizar pruebas en un sistema operativo Linux.	<ul style="list-style-type: none">• No te sientes cómodo usando comandos de línea de comandos.• No desea incurrir en ningún otro AWS cargo.• No quieres realizar pruebas en un sistema operativo Linux.
the section called “Utilice su PC o Mac con Windows o Linux como AWS IoT dispositivo” (p. 50)	<ul style="list-style-type: none">• No desea incurrir en ningún otro AWS cargo.• No desea configurar dispositivos adicionales.	<ul style="list-style-type: none">• No desea instalar ningún software en su equipo personal.• Desea una plataforma de pruebas más representativa.
the section called “Connect un Raspberry Pi u otro dispositivo” (p. 56)	<ul style="list-style-type: none">• ¿Quieres probar AWS IoT con un dispositivo real.• Ya tienes un dispositivo con el que probar.• Tiene experiencia en la integración de hardware en los sistemas.	<ul style="list-style-type: none">• No quieres comprar o configurar un dispositivo solo para probarlo.• ¿Quieres probar AWS IoT más sencillamente posible, por ahora.

CrearAWS IoTRecursos

En este tutorial creará la AWS IoTRecursos a los que necesita un dispositivo para conectarse AWS IoT para intercambiar mensajes.

Create an AWS IoT Core policy

Create a thing and its certificate



1. Creación de unAWS IoTdocumento de política, que autorizará a su dispositivo a interactuar conAWS IoTServicios de .
2. Creación de un objeto objetoAWS IoTy su certificado de dispositivo X.509 y, a continuación, adjunte el documento de política. El objeto de cosa es la representación virtual de su dispositivo en elAWS IoTregistro. El certificado autentica el dispositivo paraAWS IoT Core, y el documento de política autoriza a su dispositivo a interactuar conAWS IoT.

Note

Si planeasthe section called “Creación de un dispositivo virtual con Amazon EC2” (p. 42), puede saltarse esta página y continuar conthe section called “Configuración del dispositivo” (p. 42). Creará estos recursos cuando cree su cosa virtual.

En este tutorial se utilizaAWS IoTconsola para crear elAWS IoTde AWS. Si el dispositivo admite un navegador web, podría resultar más fácil ejecutar este procedimiento en el navegador web del dispositivo porque podrá descargar los archivos de certificado directamente en su dispositivo. Si ejecuta este procedimiento en otro equipo, tendrá que copiar los archivos de certificado en su dispositivo antes de que la aplicación de ejemplo los pueda utilizar.

Creación de una política de AWS IoT

Los dispositivos utilizan un certificado X.509 para autenticarse conAWS IoT Core. El certificado tieneAWS IoTpolíticas adjuntas a él. Estas políticas determinan quéAWS IoToperaciones de, como suscribirse a temas MQTT o publicar en ellos, se permite ejecutar el dispositivo. Su dispositivo presenta su certificado cuando se conecta y envía mensajes aAWS IoT Core.

Siga los pasos para crear una política que permita a su dispositivo ejecutar laAWS IoToperaciones necesarias para ejecutar el programa de ejemplo. Debe crear elAWS IoTantes de poder adjuntarlo al certificado de dispositivo, que creará más adelante.

Para crear una política de AWS IoT

1. En el menú de la izquierda, elijaSeguridady luego seleccionePolíticas. En la páginaTodavía no tiene una póliza, elijaCrear política.

Si tu cuenta tiene políticas existentes, eligeCrear.

2. En la páginaCrear políticaapágina:

1. En el navegadorPropiedades de las políticasen el apartadoNombre de la política, escriba un nombre para la política (por ejemplo,**My_Iot_Policy**). No utilice información personalmente identificable en sus nombres de política.
2. En el navegadorDocumento de política, cree las declaraciones de política que otorgan o deniegan el acceso a los recursos aAWS IoT Coreoperaciones. Para crear una declaración de política que conceda a todos los clientes realizar**iot:Connect**, siga estos pasos:
 - En el navegadorEfecto de políticacampo, elijaPermitir. Esto permite a todos los clientes que tienen esta política asociada a su certificado realizar la acción que se indica en elAcciones de política de.
 - En el navegadorAcciones de política de, elija una acción de política como*iot:Connect*. Las acciones de política son las acciones que el dispositivo necesita permiso para realizar cuando ejecuta el programa de ejemplo desde el SDK de dispositivos.
 - En el navegadorRecurso de políticas, escriba un recurso Amazon Resource Name (ARN) o*. UNA*para seleccionar cualquier cliente (dispositivo).

Para crear las declaraciones de política para*iot:Receive, iot:Publish, y iot:Subscribe*, eligeAdición de una nueva declaracióny repite los pasos.

Policy effect	Policy action	Policy resource	
Allow	iot:Connect	*	Remove
Allow	iot:Receive	*	Remove
Allow	iot:Publish	*	Remove
Allow	iot:Subscribe	*	Remove

Note

En este inicio rápido, el carácter comodín (*) se utiliza para simplificar. Para una mayor seguridad, debe restringir qué clientes (dispositivos) pueden conectarse y publicar mensajes, especificando un ARN de cliente en lugar del carácter comodín como recurso.

Los ARN de cliente tienen este formato:`arn:aws:iot:<your-region>:<your-aws-account>:client/<my-client-id>`.

Sin embargo, primero debe crear el recurso (como un dispositivo cliente o una sombra de cosas) para poder asignar su ARN a una política. Para obtener más información, consulte[AWS IoT CoreRecursos de acción de](#).

3. Una vez que haya especificado la información de su política, seleccioneCrear.

Para obtener más información, consulte [Políticas administradas de IAM \(p. 406\)](#).

Crear objeto objeto objeto

Dispositivos conectados aAWS IoTestán representados porObjetosen laAWS IoTregistro. UNAobjeto thingrepresenta un dispositivo específico o una entidad lógica. Puede ser un dispositivo físico o un sensor (por ejemplo, una bombilla o un interruptor de luz en la pared). También puede ser una entidad lógica, como una instancia de una aplicación o una entidad física que no se conecta conAWS IoT, pero está

relacionada con otros dispositivos que sí lo están (por ejemplo, un automóvil con sensores de motor o un panel de control).

Para crear un objeto en elAWS IoTconsola

1. En el navegador[AWS IoTconsola](#), en el menú de la izquierda, elijaManejar, luego elijaObjetos.
2. En la páginaObjetos, elijaCreación de objetos.
3. En la páginaCreación de objetos, elijaCrear un solo objeto, luego elijaPróximo.
4. En la páginaEspecificarr propiedades de cosa, paraNombre de cosa, escriba un nombre para su objeto, como**MyIotThing**.

Al nombrar cosas, elija el nombre cuidadosamente, porque no puede cambiar el nombre de una cosa después de crearlo.

Para cambiar el nombre de un objeto, debe crear otro objeto nueva, asignarle el nuevo nombre y eliminar después el objeto anterior.

Note

No utilice información personalmente identificable en su nombre de objeto. El nombre de la cosa puede aparecer en comunicaciones e informes sin cifrar.

5. Mantenga vacío el resto de los campos de esta página. Elija Next (Siguiente).
6. En la páginaConfiguración del certificado del dispositivo -opcional, elijaGenerar automáticamente un nuevo certificado (recomendado). Elija Next (Siguiente).
7. En la páginaAdjuntar políticas al certificado -opcional, seleccione la política que ha creado en la sección anterior. En esa sección, se denominó la política,**My_Iot_Policy**. ElegirCreación de objetos.
8. En la páginaDescargar certificados y clavespágina:
 1. Descargue cada uno de los archivos de certificado y clave y guárdelos para más adelante. Tendrás que instalar estos archivos en tu dispositivo.

Cuando guarde los archivos de certificado, indique los nombres de la siguiente tabla. Estos son los nombres de archivo utilizados en ejemplos posteriores.

Nombres de archivo de certificado

Archivos	Ruta de archivo
Clave privada	<code>private.pem.key</code>
Clave pública	(no se utiliza en estos ejemplos)
Certificado de dispositivo	<code>device.pem.crt</code>
Certificado de entidad de certificación raíz	<code>Amazon-root-CA-1.pem</code>

2. Descargue el archivo CA raíz de estos archivos eligiendo laDescargarenlace del archivo de certificado de CA raíz que corresponde al tipo de endpoint de datos y conjunto de cifrado que está utilizando. En este tutorial, elijaDescargara la derecha deClave RSA de 2048 bits: Amazon Root CA 1y descargueClave RSA de 2048 bits: Amazon Root CA 1archivo de certificado.

Important

Debe guardar los archivos de certificado antes de salir de esta página. Después de dejar esta página en la consola, ya no tendrá acceso a los archivos de certificado.

Si olvidó descargar los archivos de certificado que creó en este paso, debe salir de la pantalla de la consola, ir a la lista de cosas de la consola, eliminar el objeto de cosa que creó y, a continuación, reiniciar este procedimiento desde el principio.

3. Seleccione Done (Listo).

Después de completar este procedimiento, debería ver el nuevo objeto de cosa en la lista de cosas.

Configuración del dispositivo

En esta sección se describe cómo configurar el dispositivo para conectarse aAWS IoT. Si desea empezar conAWS IoTpero aún no tienes un dispositivo, puedes crear un dispositivo virtual utilizando Amazon EC2 o puedes usar tu PC Windows o Mac como dispositivo IoT.

Selecciona la mejor opción de dispositivo para que pruebesAWS IoT. Por supuesto, puedes probarlos todos, pero prueba solo uno a la vez. Si no está seguro de qué opción de dispositivo es la mejor para usted, lea sobre cómo elegir[qué opción de dispositivo es la mejor \(p. 38\)](#)y, a continuación, vuelva a esta página.

Opciones de dispositivos

- [Creación de un dispositivo virtual con Amazon EC2 \(p. 42\)](#)
- [Utilice su PC o Mac con Windows o Linux comoAWS IoTdispositivo \(p. 50\)](#)
- [Connect un Raspberry Pi u otro dispositivo \(p. 56\)](#)

Creación de un dispositivo virtual con Amazon EC2

En este tutorial, creará una instancia de Amazon EC2 para que funcione como dispositivo virtual en la nube.

Para completar este tutorial, necesita unCuenta de AWS. Si no dispone de una, complete los pasos descritos en[Configurar suCuenta de AWS \(p. 18\)](#)Antes de continuar.

En este tutorial, hará lo siguiente:

- [Configuración de una instancia de Amazon EC2 \(p. 42\)](#)
- [Instale Git, Node.js y configure elAWS CLI \(p. 43\)](#)
- [CrearAWS IoTrecursos para su dispositivo virtual \(p. 45\)](#)
- [Instalación de AWS IoT Device SDK para JavaScript \(p. 48\)](#)
- [Ejecute la aplicación de ejemplo \(p. 48\)](#)
- [Ver los mensajes de la aplicación de ejemplo en elAWS IoTconsola \(p. 50\)](#)

Configuración de una instancia de Amazon EC2

En los siguientes pasos se muestra cómo crear una instancia de Amazon EC2 que actuará como dispositivo virtual en lugar de un dispositivo físico.

Si es la primera vez que crea una instancia de Amazon EC2, puede encontrar las instrucciones en[Introducción a las instancias Amazon EC2Linux](#)para ser más útil.

Para lanzar una instancia

1. Abra la consola de Amazon EC2 en <https://console.aws.amazon.com/ec2/>.
2. En el panel de la consola, elija Launch Instance.
3. LaPaso 1: Elegir una imagen de máquina de Amazon (AMI)muestra una lista de configuraciones básicas, denominadasImágenes de Amazon Machine (AMI), que sirven de plantillas para tu instancia. Seleccione una versión HVM de Amazon Linux 2,tales comoAmazon Linux 2 AMI (HVM), SSD volumen. Observe que estas AMI están marcadas como "Free tier eligible" (Apta para el nivel gratuito).

4. En la página Choose an Instance Type (Elegir un tipo de instancia), puede seleccionar la configuración de hardware de la instancia. Seleccione el tipo `t2.micro`, que es la opción predeterminada. Observe que este tipo de instancia es apta para la capa gratuita.
5. Elija Review and Launch (Revisar y lanzar) para dejar que el asistente rellene los demás valores de configuración automáticamente.
6. En la página Review Instance Launch, elija Launch.
7. Cuando le pidan un key pair, seleccione Creación de un nuevo key pair, escriba un nombre para el key pair y después seleccione Descargar Par de claves. Esta es la única oportunidad que tiene de guardar el archivo de clave privada, así que asegúrese de descargarlo. Guarde el archivo de clave privada en un lugar seguro. Deberá proporcionar el nombre de su par de claves al lanzar una instancia, y la clave privada correspondiente cada vez que se conecte a dicha instancia.

Warning

No seleccione la opción Proceed without a key pair. Si lanza la instancia sin un par de claves, no podrá conectarse a ella.

Cuando esté preparado, elija Lanzar instancias.

8. Verá una página de confirmación que indicará que la instancia se está lanzando. Elija View Instances para cerrar la página de confirmación y volver a la consola.
9. Puede ver el estado del lanzamiento en la pantalla Instances (Instancias). La instancia tarda poco tiempo en lanzarse. Al lanzar una instancia, su estado inicial es pending. Una vez iniciada la instancia, el estado cambia a running y recibe un nombre de DNS público. (Si la columna Public DNS (IPv4) (DNS público (IPv4)) está oculta, elija Show/Hide Columns (Mostrar/ocultar columnas) (el ícono con forma de engranaje) en la esquina superior derecha de la página y, a continuación, seleccione Public DNS (IPv4) (DNS público (IPv4))).
10. Puede que transcurran unos minutos hasta que la instancia esté lista para conectarse a ella. Compruebe que la instancia haya superado las comprobaciones de estado; puede ver esta información en la columna Status Checks.

Una vez que la nueva instancia haya superado sus comprobaciones de estado, continúe con el siguiente procedimiento y conéctese a ella.

Para conectarse a la instancia

Puede conectarse a una instancia mediante el cliente basado en navegador seleccionado la instancia desde la consola de Amazon EC2 y eligiendo conectarse mediante Amazon EC2 Instance Connect. Instance Connect gestiona los permisos y proporciona una conexión exitosa.

1. Abra la consola de Amazon EC2 en <https://console.aws.amazon.com/ec2/>.
2. En el menú de la izquierda, elija Instancias.
3. Seleccione la instancia y elija Connect (Conectar).
4. Elija Amazon EC2 Instance Connect (conexión SSH basada en navegador), Conectar.

Ahora debería tener una conexión de la instancia Amazon EC2 ventana que ha iniciado sesión en su nueva instancia Amazon EC2.

Instale Git, Node.js y configure el AWS CLI

En esta sección, instalarás Git y Node.js en tu instancia de Linux.

Para instalar Git

1. En la terminal de la instancia Amazon EC2, actualice la instancia mediante el siguiente comando.

```
sudo yum update -y
```

2. En las deConexión de la instancia Amazon EC2, instale Git utilizando el siguiente comando.

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

Para instalar Node.js

1. En las deConexión de la instancia Amazon EC2window, instale el administrador de versiones de nodos (nvm) mediante el siguiente comando.

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

Utilizaremos nvm para instalar Node.js, ya que nvm puede instalar varias versiones de Node.js y permitirle alternar entre ellas.

2. En las deConexión de la instancia Amazon EC2, active nvm mediante este comando.

```
. ~/.nvm/nvm.sh
```

3. En las deConexión de la instancia Amazon EC2, utilice nvm para instalar la última versión de Node.js mediante este comando.

```
nvm install node
```

Si instala Node.js también instalará el administrador de paquetes de nodos (npm) para poder instalar módulos adicionales según sea necesario.

4. En las deConexión de la instancia Amazon EC2, compruebe que Node.js está instalado y se ejecuta correctamente utilizando este comando.

```
node -v
```

Este tutorial requiere Node v10.0 o posterior.

Para configurar AWS CLI

Su instancia Amazon EC2 viene precargada con elAWS CLI. Sin embargo, debe completar suAWS CLIperfil. Para obtener más información acerca de cómo configurar la CLI, consulte[Configuración deAWS CLI](#).

1. En el ejemplo siguiente se muestran los valores de ejemplo. Sustituya los por sus propios valores. Puede encontrar estos valores en su[AWSconsola en la información de tu cuenta enCredenciales de seguridad](#).

En las deConexión de la instancia Amazon EC2, escriba este comando:

```
aws configure
```

A continuación, introduzca los valores de su cuenta en las indicaciones que se muestran.

```
AWS Access Key ID [None]: AKIAIOSFODNN7EXAMPLE
AWS Secret Access Key [None]: wJalrXUtnFEMI/K7MDENG/bPxRfCYEXAMPLEKEY
Default region name [None]: us-west-2
```

```
Default output format [None]: json
```

2. Puedes probar tu AWS CLI configuración con este comando:

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

Si las recetas AWS CLI están configurado correctamente, el comando debe devolver una dirección de endpoint desde su cuenta de AWS.

Crear AWS IoT recursos para su dispositivo virtual

En esta sección se describe cómo utilizar la AWS CLI para crear el objeto de cosa y sus archivos de certificado directamente en el dispositivo virtual. Esto se hace directamente en el dispositivo para evitar la posible complicación que podría surgir al copiarlos en el dispositivo desde otro equipo.

Para crear un AWS IoT objeto de cosa en la instancia de Linux

Dispositivos conectados a AWS IoT están representados por objetos en la AWS IoT registro. UNA objeto thing representa un dispositivo específico o una entidad lógica. En este caso, a su objeto thing representará su dispositivo virtual, esta instancia Amazon EC2.

1. En la conexión de la instancia Amazon EC2, ejecute el siguiente comando para crear el objeto thing.

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

2. La respuesta JSON debería tener el siguiente aspecto:

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

Para crear y adjuntar AWS IoT claves y certificados en la instancia de Linux

La [create-keys-and-certificate](#) crea certificados de cliente firmados por la entidad emisora de certificados Amazon Root. Este certificado se utiliza para autenticar la identidad del dispositivo virtual.

1. En la conexión de la instancia Amazon EC2, cree un directorio para almacenar su certificado y sus archivos de clave.

```
mkdir ~/certs
```

2. En la conexión de la instancia Amazon EC2, descargue una copia del certificado de entidad de certificación (CA) de Amazon mediante este comando.

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

3. En la conexión de la instancia Amazon EC2, ejecute el siguiente comando para crear archivos de clave privada, clave pública y certificado X.509. Este comando también registra y activa el certificado con AWS 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"
```

La respuesta tiene este aspecto: Guarde el iconocertificateArnpara que puedas utilizarlo en los comandos posteriores. Lo necesitarás para adjuntar el certificado a tu cosa y para adjuntar la política al certificado en unos pasos posteriores.

```
{
    "certificateArn": "arn:aws:iot:us-
west-2:123456789012:cert/9894ba17925e663f1d29c23af4582b8e3b7619c31f3fdb93adcb51ae54b83dc2",
    "certificateId": "9894ba17925e663f1d29c23af4582b8e3b7619c31f3fdb93adcb51ae54b83dc2",
    "certificatePem": "
-----BEGIN CERTIFICATE-----
MIICITCCExAMPLE6m7oTRw0uXOjANBgkqhkiG9w0BAQUFADCBiDELMAKGA1UEBhMC
VVMxCzAJBgNVBAgEExAMPLEd0gYDVQHewdTZWFOdgxLMQ8wDQYDVQQKEWZBbWF6
b24xFDASBgNVBASTC0lBTSEXAMPLE2xLMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAd
BgkqhkiG9w0BCQEWEg5vb25lQGFTyEXAMPLEb20wHcNMTEwNDI1MjA0NTIxWhcN
MTIwNDI0MjA0NTIxWjCBiDELMAKGA1UEBhMCEXAMPLEJBgNVBAgTAldBMRAwDgYD
VQHewdTZWFOdgxLMQ8wDOYDVQOKEwZBBWF6b24xFDAExAMPLETC0lBTSDb25z
b2xLMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAdBgkqhkiG9w0BCQEExAMPLE25lQGFT
YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMA0dn+aExAMPLE
ExAMPLEfEvySwtC2XADZ4nB+BLYgViK60CpiwsZ3G93vUEIO3IyNoH/f0wYK8m9T
rDHudUZExAMPLEG5M43q7Wgc/MbQITxOUSQv7c7ugFFDzQGBzZswY6786m86gpE
Ibb30hjZnzcwQAEExAMPLEWImm2nrAgMBAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
nUhVVxYUntneD9+h8Mg9qExAMPLEyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb
FFBjvSfpJILj00zbhNYSf6GuoEDEXAMPLEBHjJnyp3780D8uTs7fLvjx79LjSTb
NYiytVbZPQQU5Yaxu2jXnimvw3rrszlaEXAMPLE=
-----END CERTIFICATE-----\n",
    "keyPair": {
        "PublicKey": "-----BEGIN PUBLIC
KEY-----\nMIIBIjANBgkqhkExAMPLEQEFAAQCAQ8AMIIBCgKCAQEAExAMPLE1nnnyJwKSMHw4h
\nMMExAMPLEuuN/dMAS3fyce8DW/4+ExAMPLEyjmof/YVF/
gHr99VEExAMPLE5VF13\n59VK7cExAMPLE67GK+y+jikqXOgHh/xJTTwo
+sGpWExAMPLEd18x0d2ka4tCzuWExAMPLEahJbYkCPUBS8opVkr7qkExAMPLE1DR6sx2HocliOOLtu6Fkw91swQWExAMPLE
\GB3ZPrNhOpzQYVjUSTzeccyNx2ExAMPLEvp9mQOUXP6plfgxwKRX2fExAMPLEDa\nhJLXkX3rHU2xbxJSq7D
+XExAMPLEcw+LyFhI5mgFRL88eGdsAExAMPLElnI9EesG/nFQIDAQAB\-----END PUBLIC KEY-----\n",
        "PrivateKey": "-----BEGIN RSA PRIVATE KEY-----\nkey omitted for security
reasons\n-----END RSA PRIVATE KEY-----\n"
    }
}
```

- En las deConexión de la instancia Amazon EC2, adjunte su objeto thing al certificado que acaba de crear mediante el siguiente comando y elcertificateArnen la respuesta del comando anterior.

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

Si se ejecuta correctamente, este comando no muestra ningún resultado.

Para crear y asociar una política

- En las deConexión de la instancia Amazon EC2, cree el archivo de política copiando y pegando este documento de política en un archivo denominado~/policy.json.

Si no dispone de un editor Linux favorito, puede abrirnano, mediante este comando.

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

Y pegar el documento de política `parapolicy.json` en él. Ingrese `ctrl-x` para salir del nanoeditor y guarde el archivo.

Contenido del documento de política `depolicy.json`.

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

2. En las deConexión de la instancia Amazon EC2, cree su política mediante el siguiente comando.

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

Salida:

```
{  
    "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. En las deConexión de la instancia Amazon EC2, adjunte la política al certificado de su dispositivo virtual mediante el siguiente comando.

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

Si se ejecuta correctamente, este comando no muestra ningún resultado.

En este punto, ha creado para su dispositivo virtual:

- Objeto de cosa en el que representar tu dispositivo virtual AWS IoT.
- Un certificado para autenticar el dispositivo virtual.
- Documento de política para autorizar a su dispositivo virtual a Connect a AWS IoT para Publicar, Recibir y Suscribirse a mensajes.

Instalación de AWS IoT Device SDK para JavaScript

En esta sección, instalará la AWS IoT SDK de dispositivos para JavaScript, que contiene el código con el que las aplicaciones pueden utilizar para comunicarse con AWS IoT los programas de ejemplo.

Para instalar el AWS IoT Device SDK para JavaScript en la instancia de Linux

1. En la conexión de la instancia Amazon EC2, clone el AWS IoT Device SDK para JavaScript en el `aws-iot-device-sdk-js-v2` directorio de su directorio personal mediante este comando.

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

2. Vaya a la `.aws-iot-device-sdk-js-v2` directorio que creó en el paso anterior.

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

3. Utilice npm para instalar el SDK.

```
npm install
```

Ejecute la aplicación de ejemplo

Los comandos de las siguientes secciones suponen que los archivos de clave y certificado se almacenan en el dispositivo virtual como se muestra en esta tabla.

Nombres de archivo de certificado

Archivos	Ruta de archivo
Clave privada	<code>~/certs/private.pem.key</code>
Certificado de dispositivo	<code>~/certs/device.pem.crt</code>
Certificado de entidad de certificación raíz	<code>~/certs/Amazon-root-CA-1.pem</code>

En esta sección, instalará y ejecutará la `pub-sub.js` aplicación de ejemplo encontrada en el `aws-iot-device-sdk-js-v2/samples/nodedirectory` del directorio AWS IoT Device SDK for JavaScript. Esta aplicación muestra cómo un dispositivo, su instancia de Amazon EC2, utiliza la biblioteca MQTT para publicar mensajes MQTT y suscribirse a ellos. La `pub-sub.js` aplicación de muestra se suscribe a un tema, `topic_1`, publica 10 mensajes en ese tema y muestra los mensajes tal como se reciben del agente de mensajes.

Para instalar y ejecutar la aplicación de ejemplo

1. En las deConexión de la instancia Amazon EC2, vaya a la ventanaaws-iot-device-sdk-js-v2/samples/node/pub_subdirectorio que el SDK creó e instala la aplicación de ejemplo mediante estos comandos.

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

2. En las deConexión de la instancia Amazon EC2ventana, obtener*su punto final de iot-ot*desdeAWS IoTmediante este comando.

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

3. En las deConexión de la instancia Amazon EC2ventana, inserte*su punto final de iot-ot*como se indica y ejecute 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
```

La aplicación de ejemplo:

1. Se conecta alAWS Iotservicio para tu cuenta.
2. Se suscribe al tema del mensaje,*tema_1*y muestra los mensajes que recibe sobre ese tema.
3. Publica 10 mensajes sobre el tema,*tema_1*.
4. Muestra una salida similar a la siguiente:

```
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}
```

Si tiene problemas al ejecutar la aplicación de ejemplo, consulte[the section called “Solución de problemas con la aplicación de ejemplo” \(p. 64\)](#).

También puede agregar el--verbosity Debuga la línea de comandos para que la aplicación de ejemplo muestre mensajes detallados sobre lo que está haciendo. Esa información podría proporcionarle la ayuda que necesita para corregir el problema.

Ver los mensajes de la aplicación de ejemplo en elAWS IoTconsola

Puede ver los mensajes de la aplicación de ejemplo a medida que pasan por el agente de mensajes utilizando elCliente MQTTen laAWS IoTconsola.

Para ver los mensajes MQTT publicados por la aplicación de ejemplo

1. Consulte [Ver los mensajes MQTT con el cliente MQTT de AWS IoT \(p. 65\)](#). Esto le ayuda a aprender a utilizar laCliente MQTTen laAWS IoTconsola para ver mensajes MQTT a medida que pasan por el agente de mensajes.
2. Abra el iconoCliente MQTTen laAWS IoTconsola.
3. Suscríbase al tema,tema_1.
4. En las deConexión de la instancia Amazon EC2, ejecute de nuevo la aplicación de muestra y vea los mensajes en elCliente MQTTen laAWS IoTconsola.

```
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
```

Utilice su PC o Mac con Windows o Linux comoAWS IoTdispositivo

En este tutorial, configurará un equipo personal para utilizarlo conAWS IoT. Estas instrucciones son compatibles con PC y Mac con Windows y Linux. Para ello, debe instalar algún software en el equipo. Si no quieres instalar software en tu equipo, puedes intentar[Creación de un dispositivo virtual con Amazon EC2 \(p. 42\)](#), que instala todo el software en una máquina virtual.

En este tutorial, hará lo siguiente:

- [Configurar tu ordenador personal \(p. 50\)](#)
- [Instale Git, Python yAWS IoTDevice SDK for Python \(p. 50\)](#)
- [Configurar la política y ejecutar la aplicación de ejemplo \(p. 53\)](#)
- [Ver los mensajes de la aplicación de ejemplo en elAWS IoTconsola \(p. 56\)](#)

Configurar tu ordenador personal

Para completar este tutorial, necesitas un PC Windows o Linux o un Mac con conexión a Internet.

Antes de continuar con el paso siguiente, asegúrese de abrir una ventana de línea de comandos en el equipo. Usarcmd.exe en un PC con Windows. En un PC Linux o Mac, utiliceTerminal.

Instale Git, Python yAWS IoTDevice SDK for Python

En esta sección, instalará Python yAWS IoTDevice SDK for Python on the computer.

Instale la versión más reciente de Git y Python

Para descargar e instalar Git y Python en el equipo

1. Compruebe si tiene Git instalado en su equipo. Introduzca este comando en la línea de comandos.

```
git --version
```

Si el comando muestra la versión de Git, Git está instalado y puedes continuar con el siguiente paso.

Si el comando muestra un error, abra <https://git-scm.com/download> e instala Git para tu ordenador.

2. Compruebe si ya ha instalado Python. Introduzca este comando en la línea de comandos.

```
python -v
```

Note

Si este comando produce un error: `Python was not found`, puede deberse a que su sistema operativo llama al ejecutable de Python v3.x como `Python3`. En ese caso, sustituya todas las instancias de `python` con `python3` y continúe con el resto de este tutorial.

Si el comando muestra la versión de Python, Python ya está instalado. Este tutorial requiere Python v3.5 o posterior.

3. Si Python está instalado, puede omitir el resto de los pasos de esta sección. Si no, siga aquí.
4. Abierto <https://www.python.org/downloads/> y descargue el instalador para su equipo.
5. Si la descarga no comenzó a instalarse automáticamente, ejecute el programa descargado para instalar Python.
6. Verifique la instalación de Python.

```
python -v
```

Confirme que el comando muestra la versión de Python. Si no se muestra la versión de Python, intente descargar e instalar Python de nuevo.

Instalación del AWS IoT Device SDK for Python

Para instalar el AWS IoT Device SDK for Python on the computer

1. Instale v2 del AWS IoT Device SDK for Python.

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

2. Clone the AWS IoT Device SDK for Python en el directorio `aws-iot-device-sdk-python-v2` del directorio principal. Este procedimiento hace referencia al directorio base de los archivos que está instalando como `hogar`.

La ubicación real del `hogar` depende de su sistema operativo.

Linux/macOS

En macOS y Linux, el `hogar` directory es ~.

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

Windows

En Windows, puede encontrar el `hogar` ruta de directorio ejecutando este comando en el cmdventana.

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

Note

Si utiliza Windows PowerShell en contraposición acmd.exe, a continuación, utilice el siguiente comando.

```
echo $home
```

Preparación antes de ejecutar las aplicaciones de ejemplo

Para preparar el sistema para ejecutar la aplicación de ejemplo

- Cree el `certs` directorio. En el `certs`, copie los archivos de clave privada, certificado de dispositivo y certificado de entidad de certificación raíz que guardó al crear y registrar el objeto de cosa en [the section called “CrearAWS IoT Recursos” \(p. 38\)](#). Los nombres de archivo de cada archivo del directorio de destino deben coincidir con los de la tabla.

Los comandos de la siguiente sección suponen que los archivos de clave y certificado se almacenan en el dispositivo como se muestra en esta tabla.

Linux/macOS

Ejecute este comando para crear el `certs` subdirectorio que utilizará cuando ejecute las aplicaciones de ejemplo.

```
mkdir ~/certs
```

En el nuevo subdirectorio, copie los archivos en las rutas de archivo de destino que se muestran en la tabla siguiente.

Nombres de archivo de certificado

Archivos	Ruta de archivo
Clave privada	<code>~/certs/private.pem.key</code>
Certificado de dispositivo	<code>~/certs/device.pem.crt</code>
Certificado de entidad de certificación raíz	<code>~/certs/Amazon-root-CA-1.pem</code>

Ejecute este comando para enumerar los archivos de `certs` y compáralos con los enumerados en la tabla.

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

Windows

Ejecute este comando para crear el `certs` subdirectorio que utilizará cuando ejecute las aplicaciones de ejemplo.

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

En el nuevo subdirectorio, copie los archivos en las rutas de archivo de destino que se muestran en la tabla siguiente.

Nombres de archivo de certificado

Archivos	Ruta de archivo
Clave privada	%USERPROFILE%\certs\\private.pem.key
Certificado de dispositivo	%USERPROFILE%\certs\device.pem.crt
Certificado de entidad de certificación raíz	%USERPROFILE%\certs\Amazon-root-CA-1.pem

Ejecute este comando para enumerar los archivos de *lacerts*y compáralos con los enumerados en la tabla.

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

Configurar la política y ejecutar la aplicación de ejemplo

En esta sección, configurará su política y ejecutará el `pubsub.py`script de ejemplo que se encuentra en el `aws-iot-device-sdk-python-v2/samples`directorío del directorioAWS IoT Device SDK para Python. Este script muestra cómo su dispositivo utiliza la biblioteca MQTT para publicar mensajes MQTT y suscribirse a ellos.

`lapubsub.py`aplicación de muestra se suscribe a un tema,`test/topic`, publica 10 mensajes en ese tema y muestra los mensajes tal como se reciben del agente de mensajes.

Para ejecutar `lapubsub.py`script de ejemplo, necesita la siguiente información:

Valores de parámetros de aplicación

Parámetro	Dónde encontrar el valor
<i>su punto final de iot-ot</i>	1. En el navegador AWS IoTconsola , en el menú de la izquierda, elijaConfiguración. 2. En la páginaConfiguración, el punto de enlace se muestra en laPunto final de datos de dispositivosección.

La*su punto final de iot-ot*value tiene un formato de:`endpoint_id-ats.iot.region.amazonaws.com`, por ejemplo,`a3qj468EXAMPLE-ats.iot.us-west-2.amazonaws.com`.

Antes de ejecutar el script, asegúrese de que la política de su cosa proporcione permisos para que el script de ejemplo se conecte, se suscriba, publique y reciba.

Para buscar y revisar el documento de política de un recurso de cosa

1. En el navegador[AWSconsola](#), en elObjetoslista, busca el recurso de cosa que representa tu dispositivo.
2. Elija el iconoNombreenlace del recurso de cosa que representa su dispositivo para abrir elDetalles de la objeto(Se ha creado el certificado).

3. En el navegadorDetalles de la objeto, en laCertificados, elija el certificado que se adjunta al recurso de cosa. Solo debe haber un certificado en la lista. Si hay más de uno, elija el certificado cuyos archivos están instalados en su dispositivo y que se utilizará para conectarse aAWS IoT.
4. En el navegadorCertificado página de detalles, en laPolíticas, seleccione la política asociada al certificado. Solo debería haber uno. Si hay más de uno, repita el siguiente paso para cada uno para asegurarse de que al menos una política conceda el acceso necesario.
5. En el navegadorPolítica depágina de descripción general, busque el editor JSON y elijaEditar documento de políticapara revisar y editar el documento de política según sea necesario.
6. La directiva JSON se muestra en el siguiente ejemplo. En el navegador"Resource"elemento, sustituir`region:account`con las deRegión de AWSyCuenta de AWSen cada uno de losResourcevalores.

```
{  
    "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 ejecutar el script de ejemplo en Linux/macOS

1. En la ventana de la línea de comandos, vaya hasta la~/aws-iot-device-sdk-python-v2/samples/node/pub_subdirectorio creado por el SDK mediante estos comandos.

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

2. En la ventana de la línea de comandos, sustituya*su punto final de iot-ot*como se indica y ejecute 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 ejecutar la aplicación de muestra en un PC con Windows

1. En la ventana de la línea de comandos, vaya hasta la%USERPROFILE%\aws-iot-device-sdk-python-v2\samples directorio que el SDK creó e instala la aplicación de ejemplo mediante estos comandos.

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

2. En la ventana de la línea de comandos, sustituya*su punto final de iot-ot* como se indica y ejecute 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
```

El script de ejemplo:

1. Se conecta alAWS IoT servicio para tu cuenta.
2. Se suscribe al tema del mensaje,prueba/temay muestra los mensajes que recibe sobre ese tema.
3. Publica 10 mensajes sobre el tema,prueba/tema.
4. Muestra una salida similar a la siguiente:

```
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}
```

Si tiene problemas al ejecutar la aplicación de ejemplo, consulte [the section called “Solución de problemas con la aplicación de ejemplo” \(p. 64\)](#).

También puede agregar el--verbosity Debuga la línea de comandos para que la aplicación de ejemplo muestre mensajes detallados sobre lo que está haciendo. Esa información puede ayudarle a corregir el problema.

Ver los mensajes de la aplicación de ejemplo en elAWS IoTconsola

Puede ver los mensajes de la aplicación de ejemplo a medida que pasan por el agente de mensajes utilizando elCliente MQTTen laAWS IoTconsola.

Para ver los mensajes MQTT publicados por la aplicación de ejemplo

1. Consulte [Ver los mensajes MQTT con el cliente MQTT de AWS IoT \(p. 65\)](#). Esto le ayuda a aprender a utilizar laCliente MQTTen laAWS IoTconsola para ver mensajes MQTT a medida que pasan por el agente de mensajes.
2. Abra el iconoCliente MQTTen laAWS IoTconsola.
3. Suscríbase al tema, prueba/tema.
4. En la ventana de línea de comandos, ejecute de nuevo la aplicación de ejemplo y vea los mensajes en laCliente MQTTen laAWS IoTconsola.

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
```

Conecta un Raspberry Pi u otro dispositivo

En esta sección, configuraremos una Raspberry Pi para utilizarla conAWS IoT. Si tienes otro dispositivo que te gustaría conectar, las instrucciones de Raspberry Pi incluyen referencias que pueden ayudarte a adaptar estas instrucciones a tu dispositivo.

Normalmente, esto tarda unos 20 minutos, pero puede tardar más si tiene que instalar muchas actualizaciones de software del sistema.

En este tutorial, hará lo siguiente:

- [Configuración del dispositivo \(p. 57\)](#)
- [Instale las herramientas y bibliotecas necesarias para elAWS IoTSDK de dispositivos \(p. 57\)](#)
- [InstalarAWS IoTSDK de dispositivos \(p. 58\)](#)
- [Instale y ejecute la aplicación de muestra \(p. 60\)](#)
- [Ver los mensajes de la aplicación de ejemplo en elAWS IoTconsola \(p. 63\)](#)

Important

Adaptar estas instrucciones a otros dispositivos y sistemas operativos puede ser un desafío. Tendrás que entender el dispositivo lo suficientemente bien como para poder interpretar estas instrucciones y aplicarlas a tu dispositivo.

Si encuentras dificultades al configurar el dispositivo paraAWS IoT, no podemos ofrecer asistencia más allá de las instrucciones de esta sección. Sin embargo, puedes probar una de las otras opciones de dispositivo como alternativa, como[Creación de un dispositivo virtual con Amazon EC2 \(p. 42\)](#)o[Utilice su PC o Mac con Windows o Linux comoAWS IoTdispositivo \(p. 50\)](#).

Configuración del dispositivo

El objetivo de este paso es recopilar lo que necesitará para configurar su dispositivo de modo que pueda iniciar el sistema operativo (SO), conectarse a Internet y permitirle interactuar con él en una interfaz de línea de comandos.

Necesitará lo siguiente para completar este tutorial:

- Una Cuenta de AWS. Si no dispone de una, complete los pasos descritos en [Configurar su cuenta de AWS \(p. 18\)](#) Antes de continuar.
- UNA [Raspberry Pi 3 Modelo B](#) o modelo más reciente. Esto podría funcionar en versiones anteriores de Raspberry Pi, pero no se han probado.
- [Raspberry Pi OS \(32 bits\)](#) o posterior. Le recomendamos que utilice la versión más reciente del sistema operativo Raspberry Pi. Las versiones anteriores del sistema operativo podrían funcionar, pero no se han probado.

Para ejecutar este ejemplo, no es necesario instalar el escritorio con la interfaz gráfica de usuario (GUI); sin embargo, si es nuevo en Raspberry Pi y su hardware Raspberry Pi lo admite, usar el escritorio con la GUI podría ser más fácil.

- Una conexión Ethernet o wifi.
- Teclado, ratón, monitor, cables, fuentes de alimentación y otro hardware requerido por el dispositivo.

Important

Antes de continuar con el paso siguiente, debe tener instalado, configurado y ejecutado su sistema operativo. El dispositivo debe estar conectado a Internet y deberá poder acceder al dispositivo mediante su interfaz de línea de comandos. El acceso a la línea de comandos puede realizarse a través de un teclado, ratón y monitor conectados directamente o mediante una interfaz remota de terminal SSH.

Si está ejecutando un sistema operativo en su Raspberry Pi que tiene una interfaz gráfica de usuario (GUI), abra una ventana de terminal en el dispositivo y lleve a cabo las siguientes instrucciones en esa ventana. De lo contrario, si te conectas a tu dispositivo mediante un terminal remoto, como PuTTY, abre un terminal remoto en tu dispositivo y úsalo.

Instale las herramientas y bibliotecas necesarias para el AWS IoT SDK de dispositivos

Antes de instalar el AWS IoT SDK del dispositivo y código de muestra, asegúrese de que su sistema esté up-to-date y cuenta con las herramientas y bibliotecas necesarias para instalar los SDK.

1. Actualizar el sistema operativo e instalar las bibliotecas necesarias

Antes de instalar el AWS IoT SDK de dispositivos, ejecute estos comandos en una ventana de terminal en su dispositivo para actualizar el sistema operativo e instalar las bibliotecas necesarias.

```
sudo apt-get update
```

```
sudo apt-get upgrade
```

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

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

2. Instale Git

Si el sistema operativo de tu dispositivo no viene con Git instalado, tendrás que instalarlo para instalar elAWS IoTDevice SDK for JavaScript.

- a. Compruebe si Git ya está instalado ejecutando este comando.

```
git --version
```

- b. Si el comando anterior devuelve la versión de Git, Git ya está instalado y puedes pasar al paso 3.
- c. Si se muestra un error al ejecutar lagit, instale Git ejecutando este comando.

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

- d. Vuelva a probar para ver si Git está instalado ejecutando este comando.

```
git --version
```

- e. Si Git está instalado, continúe con la siguiente sección. Si no es así, solucione los problemas y corrija el error antes de continuar. Necesitas que Git instale elAWS IoTDevice SDK for JavaScript.

InstalarAWS IoTSDK de dispositivos

Instalación delAWS IoTSDK de dispositivos.

Python

En esta sección, instalará Python, sus herramientas de desarrollo y elAWS IoTDevice SDK for Python (SDK) de dispositivo. Estas instrucciones son para un Raspberry Pi que ejecuta el último sistema operativo Raspberry Pi. Si tiene otro dispositivo o utiliza otro sistema operativo, es posible que tenga que adaptar estas instrucciones para su dispositivo.

1. Instale Python y sus herramientas de desarrollo

LaAWS IoT Device SDK for Python requiere que Python v3.5 o posterior se instale en Raspberry Pi.

En una ventana de terminal en su dispositivo, ejecute estos comandos.

1. Ejecute este comando para determinar la versión de Python instalada en el dispositivo.

```
python3 --version
```

Si Python está instalado, mostrará su versión.

2. Si la versión mostrada es Python 3.5 o superior, puede pasar al Paso 2.
3. Si la versión mostrada es inferior a Python 3.5, puede instalar la versión correcta ejecutando este comando.

```
sudo apt install python3
```

4. Ejecute este comando para confirmar que la versión correcta de Python ya está instalada.

```
python3 --version
```

2. Prueba para pip3

En una ventana de terminal en su dispositivo, ejecute estos comandos.

1. Ejecute este comando para ver si pip3 está instalado.

```
pip3 --version
```

2. Si el comando devuelve un número de versión, pip3 está instalado y puedes saltar al paso 3.
3. Si el comando anterior devuelve un error, ejecute este comando para instalar pip3.

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

4. Ejecute este comando para ver si pip3 está instalado.

```
pip3 --version
```

3. Instalar la versión actual de AWS IoT Device SDK for Python

Instalación del AWS IoT SDK de dispositivos para Python y descarga las aplicaciones de ejemplo en tu dispositivo.

En tu dispositivo, ejecuta estos comandos.

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

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

JavaScript

En esta sección, instalará Node.js, el gestor de paquetes npm y el AWS IoT Device SDK for JavaScript en el dispositivo. Estas instrucciones son para un Raspberry Pi que ejecuta el sistema operativo Raspberry Pi. Si tiene otro dispositivo o utiliza otro sistema operativo, es posible que tenga que adaptar estas instrucciones para su dispositivo.

1. Instale la versión más reciente de Node.js.

La AWS IoT Device SDK for JavaScript requiere que Node.js y el administrador de paquetes npm estén instalados en la Raspberry Pi.

- a. Descargue la última versión del repositorio de nodos introduciendo este comando.

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

- b. Instale Node y npm.

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

- c. Verifique la instalación de Node.

```
node -v
```

Confirme que el comando muestra la versión de nodo. Este tutorial requiere Node v10.0 o posterior. Si no se muestra la versión de Node, intente descargar de nuevo el repositorio de nodos.

- d. Verifique la instalación de npm.

```
npm -v
```

Confirme que el comando muestra la versión npm. Si no se muestra la versión npm, intente instalar Node y npm de nuevo.

- e. Reinicie el dispositivo.

```
sudo shutdown -r 0
```

Continúe después de que se reinicie el dispositivo.

2. Instalación de AWS IoT Device SDK para JavaScript

Instalación del AWS IoT Device SDK for JavaScript En su Raspberry Pi.

- a. Clone the AWS IoT Device SDK for JavaScript en el `aws-iot-device-sdk-js-v2` directorio de su `hogar` directorio. En el Raspberry Pi, el `hogar` directorio es `~/`, que se utiliza como `hogar` directorio en los siguientes comandos. Si el dispositivo utiliza una ruta distinta para el `hogar` directorio, debe reemplazar `~/` con la ruta correcta para su dispositivo en los siguientes comandos.

Estos comandos crean el `~/aws-iot-device-sdk-js-v2` y copia el código del SDK en él.

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

- b. Cambie a la `aws-iot-device-sdk-js-v2` directorio que creó en el paso anterior y ejecuten `npm install` para instalar el SDK. El comando `npm install` invocará el `crt` compilación de bibliotecas que puede tardar varios minutos en completarse.

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

Instale y ejecute la aplicación de muestra

En esta sección, instalará y ejecutará la `pubsub` aplicación de ejemplo encontrada en el AWS IoT SDK de dispositivos. Esta aplicación muestra cómo su dispositivo utiliza la biblioteca MQTT para publicar mensajes MQTT y suscribirse a ellos. La aplicación de ejemplo se suscribe a un tema, `topic_1`, publica 10 mensajes en ese tema y muestra los mensajes tal como se reciben del agente de mensajes.

Instalación de los archivos de certificado

La aplicación de ejemplo requiere que los archivos de certificado que autentican el dispositivo se instalen en el dispositivo.

Para instalar los archivos de certificado de dispositivo para la aplicación de ejemplo

1. Creación de un `certs` subdirectorio en su `hogar` ejecutando estos comandos.

```
cd ~  
mkdir certs
```

2. En el `~/certs`, copie la clave privada, el certificado de dispositivo y el certificado de entidad de certificación raíz creado anteriormente en [the section called “Crear AWS IoT Recursos” \(p. 38\)](#).

La forma de copiar los archivos de certificado en el dispositivo depende del dispositivo y del sistema operativo y no se describe aquí. Sin embargo, si el dispositivo admite una interfaz gráfica de usuario

(GUI) y tiene un navegador web, puede realizar el procedimiento descrito en the section called “CrearAWS IoTRecursos” (p. 38) desde el navegador web de tu dispositivo para descargar los archivos resultantes directamente en tu dispositivo.

Los comandos de la siguiente sección suponen que los archivos de clave y certificado se almacenan en el dispositivo como se muestra en esta tabla.

Nombres de archivo de certificado

Archivos	Ruta de archivo
Certificado de entidad de certificación raíz	~/certs/Amazon-root-CA-1.pem
Certificado de dispositivo	~/certs/certificate.pem.crt
Clave privada	~/certs/private.pem.key

Para ejecutar la aplicación de ejemplo, necesita la siguiente información:

Valores de parámetros de aplicación

Parámetro	Dónde encontrar el valor
<i>su punto final de iot-ot</i>	En el navegador AWS IoTConsola, eligeManejary luego seleccioneObjetos. Elige la cosa de IoT que has creado para tu dispositivo, Mi ponchefue el nombre utilizado anteriormente y, a continuación, elijalInteractuar. En la página de detalles de la cosa, su punto de enlace se muestra en elHTTPSsección. Si utilizas el nuevoAWS IoTConsola de, elijaConfiguracióndesde lasAWS IoTmenú. El punto de enlace se muestra en laPunto final de datos de dispositivosección.

La*su punto final de iot-ot*value tiene un formato de:*endpoint_id-ats.iot.region.amazonaws.com*, por ejemplo,a3qj468EXAMPLE-ats.iot.us-west-2.amazonaws.com.

Python

Para instalar y ejecutar la aplicación de ejemplo

1. Desplácese hasta el directorio de aplicaciones de muestra.

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

2. En la ventana de la línea de comandos, sustituya*su punto final de iot-ot*como se indica y ejecute 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 la aplicación de ejemplo:

1. Se conecta al AWS IoT servicio para tu cuenta.
2. Se suscribe al tema del mensaje, tema_1 y muestra los mensajes que recibe sobre ese tema.
3. Publica 10 mensajes sobre el tema, tema_1.
4. Muestra una salida similar a la siguiente:

```
Connecting to a3qEXAMPLEfffp-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!
```

Si tiene problemas al ejecutar la aplicación de ejemplo, consulte [the section called “Solución de problemas con la aplicación de ejemplo” \(p. 64\)](#).

También puede agregar el `--verbosity Debug` a la línea de comandos para que la aplicación de ejemplo muestre mensajes detallados sobre lo que está haciendo. Esta información podría proporcionarle la ayuda que necesita para corregir el problema.

JavaScript

Para instalar y ejecutar la aplicación de ejemplo

1. En la ventana de la línea de comandos, vaya hasta la `~/aws-iot-device-sdk-js-v2/samples/node/pub` subdirectorio que el SDK creó e instala la aplicación de ejemplo mediante estos comandos. El comando `npm install` invocará la `aws-crt` compilación de bibliotecas que puede tardar varios minutos en completarse.

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

2. En la ventana de la línea de comandos, sustituya `su punto final de iot-ot` como se indica y ejecute 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 la aplicación de ejemplo:
 1. Se conecta alAWS IoTservicio para tu cuenta.
 2. Se suscribe al tema del mensaje,tema_1y muestra los mensajes que recibe sobre ese tema.
 3. Publica 10 mensajes sobre el tema,tema_1.
 4. Muestra una salida similar a la siguiente:

```
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}
```

Si tiene problemas al ejecutar la aplicación de ejemplo, consulte[the section called “Solución de problemas con la aplicación de ejemplo” \(p. 64\)](#).

También puede agregar el--verbosity Debuga la línea de comandos para que la aplicación de ejemplo muestre mensajes detallados sobre lo que está haciendo. Esa información podría proporcionarle la ayuda que necesita para corregir el problema.

Ver los mensajes de la aplicación de ejemplo en elAWS IoTconsola

Puede ver los mensajes de la aplicación de ejemplo a medida que pasan por el agente de mensajes utilizando elCliente MQTTen laAWS IoTconsola.

Para ver los mensajes MQTT publicados por la aplicación de ejemplo

1. Consulte [Ver los mensajes MQTT con el cliente MQTT de AWS IoT \(p. 65\)](#). Esto le ayuda a aprender a utilizar laCliente MQTTen laAWS IoTconsolapara ver mensajes MQTT a medida que pasan por el agente de mensajes.
2. Abra el iconoCliente MQTTen laAWS IoTconsola.
3. Suscríbase al tema,tema_1.
4. En la ventana de línea de comandos, ejecute de nuevo la aplicación de ejemplo y vea los mensajes en laCliente MQTTen laAWS IoTconsola.

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
```

Solución de problemas con la aplicación de ejemplo

Si se produce un error al intentar ejecutar la aplicación de muestra, estas son algunas cosas que debe comprobar.

Compruebe el certificado

Si el certificado no está activo, AWS IoT no aceptará ningún intento de conexión que lo utilice como autorización. Al crear el certificado, es fácil pasar por alto el botón Activar. Afortunadamente, puede activar su certificado desde la [AWS IoT consola](#).

Para comprobar la activación de su certificado

1. En el navegador [AWS IoT consola](#), en el menú de la izquierda, elija Seguridad y luego seleccione Certificados.
2. En la lista de certificados, busque el certificado creado para el ejercicio y compruebe su estado en la columna Estado.

Si no recuerdas el nombre del certificado, comprueba si hay alguno que sea Inactivo para ver si es posible que sean el que estás usando.

Seleccione el certificado de la lista para abrir su página de detalles. En la página de detalles, puedes ver su Creación de fecha para ayudarte a identificar el certificado.

3. Para activar un certificado inactivo, en la página de detalles del certificado, elija Actions y luego elija Activate.

Si has encontrado el certificado correcto y está activo, pero sigues teniendo problemas para ejecutar la aplicación de ejemplo, comprueba su política como se describe en el siguiente paso.

También puede intentar crear una cosa nueva y un nuevo certificado siguiendo los pasos que se indican en la sección titulada “[Crear objeto](#)” (p. 40). Si creas algo nuevo, tendrás que darle un nombre de cosa nueva y descargar los nuevos archivos de certificado en tu dispositivo.

Verifique la política asociada al certificado

Las políticas autorizan acciones en AWS IoT. Si el certificado utilizado para conectarse a AWS IoT no tiene una política o no tiene una política que le permita conectarse, se rechazará la conexión, incluso si el certificado está activo.

Para comprobar las políticas adjuntas a un certificado

1. Busque el certificado tal y como se describe en el elemento anterior y abra su página de detalles.

2. En el menú izquierdo de la página de detalles del certificado, elija **Políticas** para consultar las políticas asociadas al certificado.
3. Si no hay políticas adjuntas al certificado, añada una eligiendo la **Actions** menú y, a continuación, elegir **Asociar política**.
Elija la política que creó anteriormente en [the section called “Crear AWS IoT Recursos” \(p. 38\)](#).
4. Si hay una política asociada, elija el mosaico de política para abrir la página de detalles.

En la página de detalles, consulte la [Documento de política](#) para asegurarse de que contiene la misma información que la que creó en [the section called “Creación de una política de AWS IoT” \(p. 39\)](#).

Compruebe la línea de comando

Asegúrese de utilizar la línea de comandos correcta para su sistema. Los comandos utilizados en los sistemas Linux/macOS suelen ser diferentes de los que se utilizan en sistemas Windows.

Comprobar la dirección del endpoint

Revise el comando que ha introducido y vuelva a comprobar la dirección del endpoint del comando con la de su [AWS IoT consola](#).

Comprobar los nombres de archivo de los archivos de certificado

Compare los nombres de archivo del comando que ha introducido con los nombres de archivo de los archivos de certificado en el `certs` directory.

Algunos sistemas pueden requerir que los nombres de los archivos estén entre comillas para que funcionen correctamente.

Verifique la instalación del SDK

Asegúrese de que la instalación del SDK esté completa y sea correcta.

En caso de duda, vuelve a instalar el SDK en tu dispositivo. En la mayoría de los casos, se trata de encontrar la sección del tutorial titulada [Instalación del AWS IoT Device SDK for Lenguaje SDK](#) siguiendo el procedimiento de nuevo.

Si utiliza la herramienta [AWS IoT Device SDK for JavaScript](#), recuerda instalar las aplicaciones de ejemplo antes de intentar ejecutarlas. La instalación del SDK no instala automáticamente las aplicaciones de ejemplo. Las aplicaciones de ejemplo deben instalarse manualmente después de instalar el SDK.

Ver los mensajes MQTT con el cliente MQTT de AWS IoT

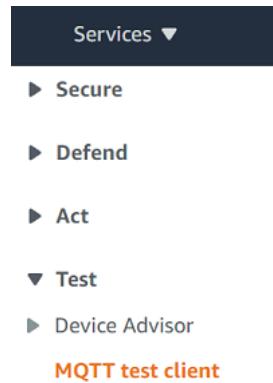
En esta sección se describe cómo utilizar la [AWS IoT Client MQTT](#) en el [AWS IoT consola](#) para ver los mensajes MQTT enviados y recibidos por AWS IoT. El ejemplo utilizado en esta sección se refiere a los ejemplos utilizados en [Introducción a AWS IoT Core \(p. 17\)](#); sin embargo, puede sustituir la `topicName` utilizado en los ejemplos con cualquier [nombre de tema o filtro de temas \(p. 98\)](#) utilizado por su solución IoT.

Los dispositivos publican los mensajes MQTT identificados por [Temas de \(p. 98\)](#) para comunicar su estado a AWS IoT, y AWS IoT publica mensajes MQTT para informar a los dispositivos y aplicaciones de cambios y eventos. Puede utilizar el cliente MQTT de para suscribirse a estos temas y ver los mensajes a medida que se producen. También puede utilizar el cliente MQTT de para publicar mensajes MQTT en dispositivos y servicios suscritos en su Cuenta de AWS.

Visualización de mensajes MQTT en el cliente MQTT

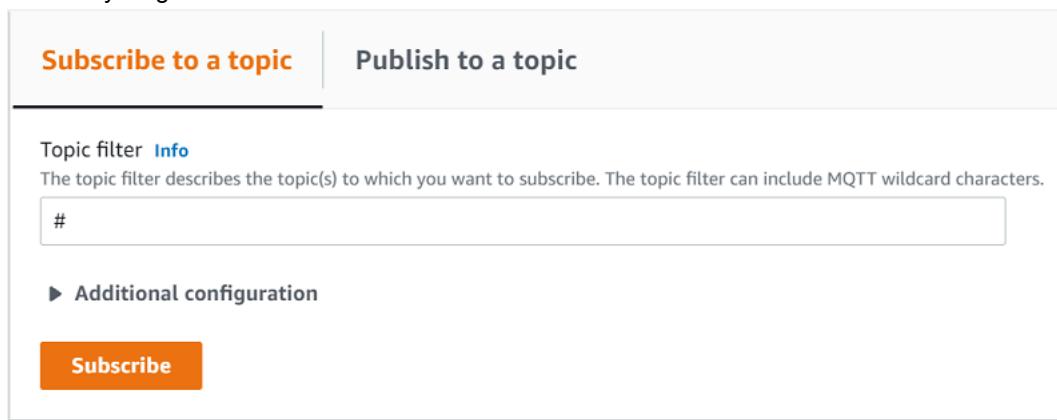
Para consultar los mensajes MQTT en el cliente MQTT

1. En el navegador[AWS IoT consola](#), en el menú de la izquierda, elija Pruebas y luego elija Cliente de pruebas MQTT.



2. En el navegador Suscripción a un tema, introduzca la pestaña *topicName* para suscribirse al tema en el que publica su dispositivo. Para obtener la aplicación de ejemplo de introducción, suscríbase a #, que se suscribe a todos los temas de los mensajes.

Continuando con el ejemplo de introducción, en la Suscripción a un tema, en la pestaña Filtro de temas, introduzca # y luego seleccione Suscribirse.



La página de registro de mensajes del tema, # se abre y aparece en el directorio Suscripciones lista. Si el dispositivo que configuró en [the section called "Configuración del dispositivo" \(p. 42\)](#) está ejecutando el programa de ejemplo, debería ver los mensajes a los que envía AWS IoT en la lista de mensajes. Las entradas del registro de mensajes aparecerán debajo de la Publicación cuando los mensajes con el tema suscrito son recibidos por AWS IoT.

Subscriptions	#	Pause	Clear	Export	Edit
#	Heartbeat X				

3. En la página #página de registro de mensajes, también puede publicar mensajes en un tema, pero tendrá que especificar el nombre del tema. No se puede publicar en el #tema.

Los mensajes publicados en los temas suscritos aparecen en el registro de mensajes a medida que se reciben, primero el mensaje más reciente.

Solución de problemas con los mensajes MQTT

Utilizar el filtro de temas comodín

Si los mensajes no aparecen en el registro de mensajes como espera, pruebe a suscribirse a un filtro de temas comodín como se describe en [Filtros de temas \(p. 99\)](#). El filtro de temas comodín multinivel MQTT es el signo hash o libra (#) y se puede utilizar como filtro de temas en el `Subscription topic`.

Suscripción a la #el filtro de temas se suscribe a todos los temas recibidos por el agente de mensajes. Puede reducir el filtro reemplazando los elementos de la ruta de filtro del tema por un #carácter comodín multinivel o el carácter comodín de un solo nivel '+'.

Al utilizar comodines en un filtro de temas

- El carácter comodín multinivel debe ser el último carácter del filtro de temas.
- La ruta del filtro de temas solo puede tener un carácter comodín de un solo nivel por nivel de tema.

Por ejemplo:

Filtro de temas	Muestra mensajes con
#	Nombre de cualquier tema
topic_1/#	Un nombre de tema que empieza por <code>topic_1/</code>
topic_1/level_2/#	Un nombre de tema que empieza por <code>topic_1/level_2/</code>
topic_1/+/level_3	Un nombre de tema que empieza por <code>topic_1/</code> , termina con <code>/level_3</code> , y tiene un elemento de cualquier valor intermedio.

Para obtener más información sobre filtros de temas, consulte [Filtros de temas \(p. 99\)](#).

Comprobar errores de nombre de tema

Los nombres de temas MQTT y los filtros de temas distinguen entre mayúsculas Si, por ejemplo, el dispositivo publica mensajes en `Topic_1`(con mayúscula T) en lugar de `topic_1`, tema al que se ha suscrito, sus mensajes no aparecerán en el cliente MQTT. Sin embargo, al suscribirse al filtro de temas comodín, se mostrará que el dispositivo está publicando mensajes y podría ver que estaba usando un nombre de tema que no era el que esperaba.

Publicación de mensajes MQTT desde el cliente MQTT

Para publicar un mensaje en un tema de MQTT

1. En la página Cliente de MQTT, en la `Publicación de un tema`, en la pestaña `Nombre del tema`, introduzca `topicName` de su mensaje. En este ejemplo, use `my/topic`.

Note

No utilice información de identificación personal en nombres de temas, ya sea que los utilice en el cliente de MQTT o en la implementación de su sistema. Los nombres de los temas pueden aparecer en comunicaciones e informes sin cifrar.

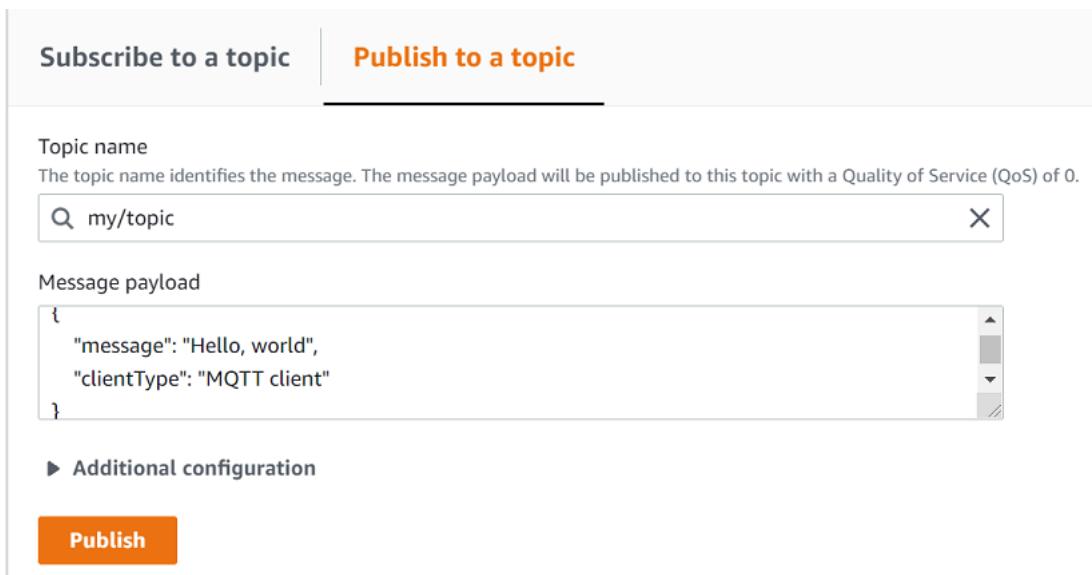
2. En la ventana de carga del mensaje, escriba el siguiente JSON:

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

3. ElegirPublicaciónpara publicar el mensaje enAWS IoT.

Note

Asegúrese de que está suscrito a lami/topic tema antes de publicar el mensaje.



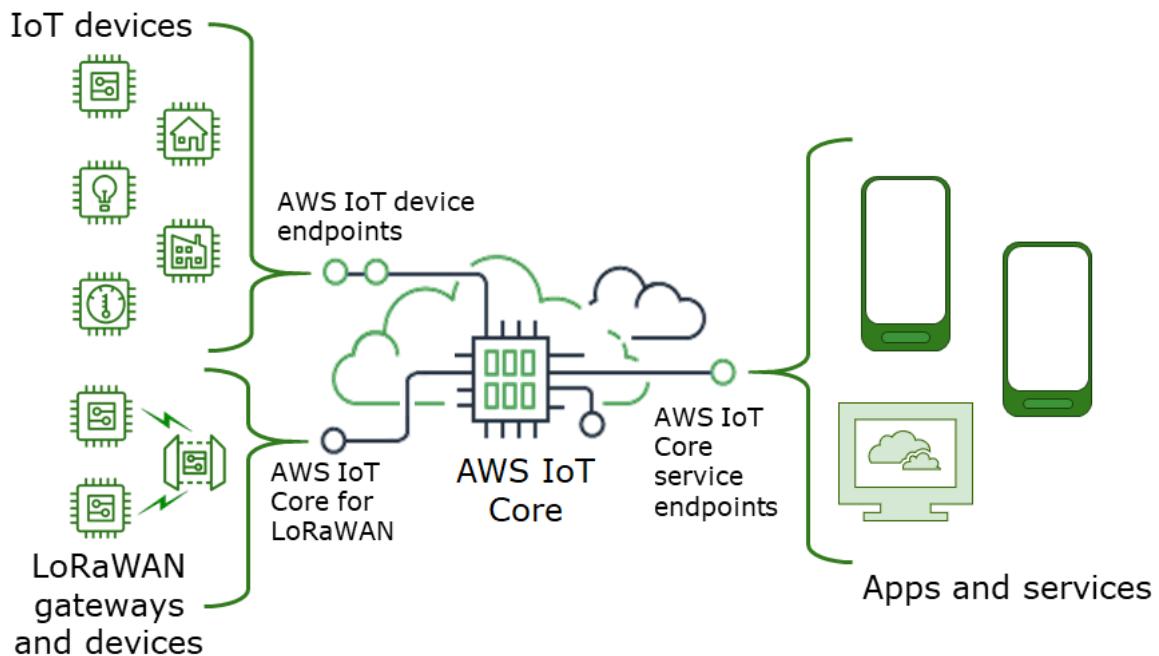
4. En el navegadorSuscripcioneslist, elijami/topicpara ver el mensaje. Debería ver el mensaje aparecer en el cliente MQTT debajo de la ventana de carga útil de mensajes de publicación.

Subscriptions	#	Pause	Clear	Export	Edit	
#	Heart	X				
	▼ my/topic		November 02, 2021, 11:55:22 (UTC-0700)			
	{ "message": "Hello, world", "clientType": "MQTT client" }					

Puede publicar mensajes MQTT en otros temas cambiando la `topicName` en la Nombre del tema y elegir el Publicación botón.

Conexión a AWS IoT Core

AWS IoT Core admite conexiones con dispositivos IoT, puertas de enlace inalámbricas, servicios y aplicaciones. Los dispositivos se conectan al AWS IoT Core para que puedan enviar datos a y recibir datos de AWS IoT servicios y otros dispositivos. Las aplicaciones y otros servicios también se conectan a AWS IoT Core para controlar y administrar los dispositivos IoT y procesar los datos de su solución IoT. En esta sección se describe cómo elegir la mejor forma de conectarse y comunicarse con AWS IoT Core para cada aspecto de la solución IoT.



Existen varias formas de interactuar con AWS IoT. Las aplicaciones y los servicios pueden utilizar el [AWS IoT Core- Puntos de enlace del plano de control \(p. 69\)](#) y los dispositivos se pueden conectar a AWS IoT Core mediante el uso de la [AWS IoT endpoints de dispositivo \(p. 70\)](#) o [AWS IoT Core para puertas de enlace y dispositivos LoRaWAN \(p. 71\)](#).

AWS IoT Core- Puntos de enlace del plano de control

La AWS IoT Core- plano de controllos endpoints proporcionan acceso a funciones que controlan y administran su AWS IoT solución.

- Puntos de enlace

La AWS IoT Core- plano de control y AWS IoT Core Plan de control de Device Advisor los puntos finales son específicos de la región y se enumeran en [AWS IoT Core Cuotas y puntos de enlace de](#). Los formatos de los endpoints son los siguientes.

Propósito del punto de enlace	Formato de punto de conexión	Sirve
AWS IoT Core- plano de control	<code>iot.aws- region.amazonaws.com</code>	AWS IoTAPI de plano de control
AWS IoT CoreDevice Advisor - plano de control	<code>api.iotdeviceadvisor.aws- region.amazonaws.com</code>	AWS IoT CoreAPI de plano de control de Device Advisor

- SDK y herramientas de

La [AWS SDK](#) proporciona soporte específico del idioma para el AWS IoT CoreAPI y API de otros AWS servicios. La [AWS Mobile SDK](#) proporciona a los desarrolladores de aplicaciones soporte específico de la plataforma para el AWS IoT CoreAPI y otros AWS servicios en dispositivos móviles.

La [AWS CLI](#) proporciona acceso de línea de comandos a las funciones proporcionadas por el AWS IoT. Puntos de enlace de servicio de . La [AWS Tools for PowerShell](#) proporciona herramientas para administrar AWS servicios y recursos del entorno de secuencias de comandos de PowerShell.

- Autenticación

Los endpoints de servicio utilizan usuarios de IAM y AWS credenciales para autenticar a los usuarios.

- Más información

Para obtener más información y vínculos a referencias del SDK, consulte [the section called “Conexión a AWS IoT Points of Service” \(p. 72\)](#).

AWS IoT endpoints de dispositivo

La AWS IoT los endpoints de dispositivos admiten la comunicación entre sus dispositivos IoT y AWS IoT.

- Puntos de enlace

Compatibilidad con los endpoints del dispositivo AWS IoT Core y AWS IoT Device Management Funciones de . Son específicos de su cuenta de AWS y puedes ver qué son usando el [describe-endpoint](#) comando.

Propósito del punto de enlace	Formato de punto de conexión	Sirve
AWS IoT Core- plano de datos	Consulte ??? (p. 78) .	AWS IoTAPI de plano de datos
AWS IoT Device Management- datos de trabajo	Consulte ??? (p. 78) .	AWS IoTAPI de plano de datos de trabajos
AWS IoT Device Advisor - plano de datos	Consulte ??? (p. 1061) .	
AWS IoT Device Management- Fleet Hub		
AWS IoT Device Management- tunelización segura	<code>api.tunneling.iot.aws- region.amazonaws.com</code>	AWS IoTAPI de tunelización segura

Para obtener más información acerca de estos puntos de enlace y las funciones que admiten, consulte [the section called “AWS IoT Data Points and Service Endpoints” \(p. 78\)](#).

- SDK

La [AWS IoT SDKs de dispositivos \(p. 80\)](#) proporciona compatibilidad específica del idioma para los protocolos de transporte de telemetría de Message Queue Server (MQTT) y WebSocket Secure (WSS), con los que los dispositivos utilizan para comunicarse con AWS IoT. [AWS Mobile SDK \(p. 76\)](#) también proporciona soporte para las comunicaciones de dispositivos MQTT, AWS IoT API y API de otros servicios en dispositivos móviles.

- Autenticación

Los endpoints del dispositivo utilizan certificados X.509 o AWS Usuarios de IAM con credenciales para autenticar a los usuarios.

- Más información

Para obtener más información y vínculos a referencias del SDK, consulte [the section called "SDK de dispositivos de AWS IoT" \(p. 80\)](#).

AWS IoT Corepara puertas de enlace y dispositivos LoRaWAN

AWS IoT Corepara LoraWAN conecta puertas de enlace y dispositivos inalámbricos a AWS IoT Core.

- Puntos de enlace

AWS IoT Corepara LoraWAN administra las conexiones de puerta de enlace a una cuenta y una región específica AWS IoT Core. Puntos de enlace de . Las puertas de enlace pueden conectarse al extremo del servidor de configuración y actualización (CUPS) de su cuenta que AWS IoT Corepara LoRaWAN proporciona.

Propósito del punto de enlace	Formato de punto de conexión	Sirve
Servidor de configuración y actualización (CUPS)	<code>account-specific-prefix.cups.lorawan.aws-region.amazonaws.com:443</code>	Comunicación de puerta de enlace con el servidor de configuración y actualización proporcionado por AWS IoT Core para LoRaWAN
Servidor de red LoRaWAN (LNS)	<code>account-specific-prefix.gateway.lorawan.aws-region.amazonaws.com:443</code>	Comunicación de puerta de enlace con el servidor de red LoRaWAN proporcionado por AWS IoT Core para LoRaWAN

- SDK

La [AWS IoT API inalámbrica](#) que AWS IoT Corepara LoRaWAN sobre el que se basa está respaldado por el [AWS SDK](#). Para obtener más información, consulte [AWS SDK y conjuntos de herramientas](#).

- Autenticación

AWS IoT Corepara comunicaciones de dispositivos LoRaWAN utilice certificados X.509 para proteger las comunicaciones con AWS IoT.

- Más información

Para obtener más información acerca de la configuración y conexión de dispositivos inalámbricos, consulte [AWS IoT Core para LoRaWAN \(p. 1116\)](#).

Conexión aAWS IoT CorePuntos de enlace de los servicios de

Puede obtener acceso a las características de laAWS IoT Core- plano de control mediante el uso de laAWS CLI, elAWSSDK para su idioma preferido o llamando directamente a la API REST. Le recomendamos que utilice laAWS CLIo unAWSSDK con el que interactuarAWS IoT Coreporque incorporan las prácticas recomendadas para llamarAWSServicios de . Llamar directamente a las API REST es una opción, pero debe proporcionar[las credenciales de seguridad necesarias](#)que permiten obtener acceso a la API.

Note

Los dispositivos IoT deben utilizar[SDK de dispositivos de AWS IoT \(p. 80\)](#). Los SDK de dispositivos están optimizados para su uso en dispositivos, admiten la comunicación MQTT conAWS IoT, y apoyar elAWS IoTAPI más utilizadas por los dispositivos. Para obtener más información acerca de los SDK de dispositivos de y las características que proporcionan, consulte[SDK de dispositivos de AWS IoT \(p. 80\)](#).

Los dispositivos móviles deben utilizar[AWS Mobile SDK \(p. 76\)](#). Los SDK móviles proporcionan soporte paraAWS IoTAPI, comunicaciones de dispositivos MQTT y API de otrosAWSservicios en dispositivos móviles. Para obtener más información acerca de los SDK de móviles y las características que proporcionan, consulte[AWS Mobile SDK \(p. 76\)](#).

Puede usarAWS Amplifyherramientas y recursos en aplicaciones web y móviles para conectarse más fácilmente aAWS IoT Core. Para obtener más información acerca de la conexión aAWS IoT Coremediante Amplify, consulte[Pub Sub Introducción](#)en la documentación de Amplify.

En las secciones siguientes se describen las herramientas y SDK que puede usar para desarrollar e interactuar con ellos.AWS IoT y otrosAWSservicios de . Para ver la lista completa de AWS herramientas y kits de desarrollo disponibles para crear y administrar aplicaciones enAWS, consulte[Herramientas sobre las que basarseAWS](#).

AWS CLI para AWS IoT Core

LaAWS CLIproporciona acceso de línea de comandos aAWSAPI.

- Instalación

Para obtener información acerca de cómo instalar laAWS CLI, consulte[Instalación de AWS CLI](#).

- Autenticación

LaAWS CLIutiliza las credenciales de suCuenta de AWS.

- Referencia

Para obtener más información sobre laAWS CLIcomandos para estosAWS IoT Coresservicios, consulte:

- [AWS CLIReferencia de comandos de IoT](#)
- [AWS CLIReferencia de comandos para datos de IoT](#)
- [AWS CLIReferencia de comandos para datos de trabajos de IoT](#)
- [AWS CLIReferencia de comandos para túneles seguros de IoT](#)

Para herramientas que administrarAWSservicios y recursos en el entorno de scripting de PowerShell, consulte[AWS Tools for PowerShell](#).

AWS SDK

con AWS Los SDK, las aplicaciones y los dispositivos compatibles pueden llamar AWS IoT API y API de otros AWS Servicios de . Esta sección proporciona enlaces a la AWSSDK y la documentación de referencia de la API de las API de la AWS IoT Core Servicios de .

La AWS Los SDK son compatibles con AWS IoT Core API de

- AWS IoT
- AWS IoT Plan de datos de
- AWS IoT Plan de datos de trabajos
- AWS IoT Tunelización segura de
- AWS IoT Wireless

C++

Para instalar el [AWS SDK for C++](#) y utilizarlo para conectarse a AWS IoT:

1. Siga las instrucciones en [Introducción al uso de la AWSSDK para C++](#)

Estas instrucciones describen cómo:

- Instalar y compilar el SDK a partir de archivos de origen
- Proporcione credenciales para utilizar el SDK con su cuenta de AWS
- Inicializar y apagar el SDK en su aplicación o servicio
- Crea un proyecto de CMake para crear tu aplicación o servicio

2. Crea y ejecuta una aplicación de ejemplo. Para aplicaciones de ejemplo que utilizan el AWSSDK for C++, consulte [AWS SDK for C++ Ejemplos de código](#).

Documentación de la para AWS IoT Coreservicios que el AWS SDK for C++ es compatible con

- [Documentación de referencia de Aws# IoT# IoTCClient](#)
- [Aws# iOTDataPlane# IOTDataPlane Documentación de referencia de cliente](#)
- [Aws# IoTJobsDataPlane# IOTJobsDataplane Documentación de referencia de cliente](#)
- [Aws# IoTSecureTunneling# IOTSecureTunneling Documentación de referencia del cliente](#)

Go

Para instalar el [AWS SDK for Go](#) y utilizarlo para conectarse a AWS IoT:

1. Siga las instrucciones en [Introducción a AWS SDK for Go](#)

Estas instrucciones describen cómo:

- Instalar la AWS SDK for Go
- Obtenga claves de acceso para que el SDK tenga acceso a su cuenta de AWS
- Importar paquetes al código fuente de nuestras aplicaciones o servicios

2. Crea y ejecuta una aplicación de ejemplo. Para aplicaciones de ejemplo que utilizan el AWS SDK for Go, consulte [AWS SDK for Go Ejemplos de código](#).

Documentación de la para AWS IoT Coreservicios que el AWS SDK for Go es compatible con

- [Documentación de referencia de IoT](#)

- [Documentación de referencia de IoT DataPlane](#)
- [Documentación de referencia de IOTJobsData Plane](#)
- [Documentación de referencia de iOTSecureTunnel](#)

Java

Para instalar el[AWS SDK for Java](#) utilizarlo para conectarse aAWS IoT:

1. Siga las instrucciones en[Introducción aAWS SDK for Java 2.x](#)

Estas instrucciones describen cómo:

- Registrarse enAWSy Creación de un usuario de IAM
- Descargar el SDK
- ConfigurarAWSCredenciales y región
- Uso del SDK con Apache Maven
- Utilizar el SDK con Gradle

2. Cree y ejecute una aplicación de ejemplo utilizando una de las[AWS SDK for Java 2.xEjemplos de código](#).

3. Consulte el[Documentación de referencia del API de SDK](#)

Documentación de la paraAWS IoT Coreservicios que elAWS SDK for Javaes compatible con

- [Documentación de referencia de IOTClient](#)
- [Documentación de referencia de cliente de IoT DataPlaneClient](#)
- [Documentación de referencia de cliente de IOTJobsDataplane](#)
- [Documentación de referencia de cliente de iOTSecureTunnelingClient](#)

JavaScript

Para instalar el[AWS SDK for JavaScript](#) utilizarlo para conectarse aAWS IoT:

1. Siga las instrucciones en[Configuración delAWS SDK for JavaScript](#). Estas instrucciones se aplican al uso de laAWS SDK for JavaScripten el navegador y con Node.JS. Asegúrese de seguir las instrucciones aplicables a la instalación.

Estas instrucciones describen cómo:

- Verifique los requisitos previos
- Instalación del SDK para JavaScript
- Carga del SDK para JavaScript

2. Cree y ejecute una aplicación de ejemplo para empezar a utilizar el SDK como describe la opción de introducción de su entorno.

- Comience a utilizar el[AWSSDK for JavaScript in the Browser](#), o bien
- Comience a utilizar el[AWSSDK de para JavaScript en Node.js](#)

Documentación delAWS IoT Coreservicios que elAWS SDK for JavaScriptes compatible con

- [AWS.Iot reference documentation](#)
- [AWS.IotData reference documentation](#)
- [AWS.IotJobsDataPlane reference documentation](#)

- [AWS.IotSecureTunneling reference documentation](#)

.NET

Para instalar el[AWS SDK for .NET](#)y utilizarlo para conectarse aAWS IoT:

1. Siga las instrucciones en[Configurar suAWS SDK for .NETEntorno](#)
2. Siga las instrucciones en[Configurar suAWS SDK for .NETProyecto de](#)

Estas instrucciones describen cómo:

- Inicie un nuevo proyecto
- Obtener y configurarAWScredenciales
- InstalarAWSPaquetes SDK

3. Cree y ejecute uno de los programas de ejemplo en[Uso deAWSservicios delAWSSDK para .NET](#)
4. Consulte el[Documentación de referencia del API de SDK](#)

Documentación delAWS IoT Coreservicios que elAWS SDK for .NETes compatible con

- [Documentación de referencia de Amazon.iot.model](#)
- [Documentación de referencia de Amazon.iotData.model](#)
- [Documentación de referencia de Amazon.iotJobsDataPlane.model](#)
- [Documentación de referencia de Amazon.iotSecureTunneling.model](#)

PHP

Para instalar el[AWS SDK for PHP](#)y utilizarlo para conectarse aAWS IoT:

1. Siga las instrucciones en[Introducción aAWS SDK for PHPVersión 3](#)

Estas instrucciones describen cómo:

- Verifique los requisitos previos
- Instalación del SDK
- Aplicar el SDK a un script PHP

2. Cree y ejecute una aplicación de ejemplo utilizando una de las[AWS SDK for PHPEjemplos de código de versión 3](#)

Documentación delAWS IoT Coreservicios que elAWS SDK for PHPes compatible con

- [Documentación de referencia de IOTClient](#)
- [Documentación de referencia de cliente de IoT DataPlaneClient](#)
- [Documentación de referencia de cliente de IOTJobsDataplane](#)
- [Documentación de referencia de cliente de iOTSecureTunnelingClient](#)

Python

Para instalar el[AWS SDK for Python \(Boto3\)](#)y utilizarlo para conectarse aAWS IoT:

1. Siga las instrucciones en la[AWS SDK for Python \(Boto3\)Inicio rápido](#)

Estas instrucciones describen cómo:

- Instalación del SDK

- Configuración del SDK
 - Utilizar el SDK de su código
2. Crear y ejecutar un programa de ejemplo que utilice la AWS SDK for Python (Boto3)

Este programa muestra las opciones de registro configuradas actualmente en la cuenta. Después de instalar el SDK y configurarlo para su cuenta, debería poder ejecutar 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 obtener más información acerca de la función utilizada en este ejemplo, consulte [the section called “Configuración de registros de AWS IoT” \(p. 427\)](#).

Documentación del AWS IoT Coreservicios que el AWS SDK for Python (Boto3) es compatible con

- [Documentación de referencia de IoT](#)
- [Documentación de referencia de IOTDataPlane](#)
- [Documentación de referencia de IOTJobsData Plane](#)
- [Documentación de referencia de iOTSecureTunnel](#)

Ruby

Para instalar el [AWS SDK for Ruby](#) y utilizarlo para conectarse a AWS IoT:

- Siga las instrucciones en [Introducción a AWS SDK for Ruby](#)
Estas instrucciones describen cómo:
 - Instalación del SDK
 - Configuración del SDK
 - Cree y ejecute el [Tutorial Hello World](#)

Documentación del AWS IoT Coreservicios que el AWSSDK for Ruby es compatible

- [Aws# IoT# Documentación de referencia de clientes](#)
- [Aws# IOTDataPlane# Documentación de referencia del cliente](#)
- [Aws# iOTJobsDataPlane# Documentación de referencia del cliente](#)
- [Aws# IoTSecureTunneling# Documentación de referencia del cliente](#)

AWS Mobile SDK

La AWS Los SDK móviles proporcionan soporte específico de la plataforma para desarrolladores de aplicaciones móviles para las API del AWS IoT Coreservicios, comunicación de dispositivos IoT mediante MQTT y las API de otros AWS Servicios de .

Android

AWS Mobile SDK for Android

La AWS Mobile SDK for Android contiene una biblioteca, ejemplos y documentación para que los desarrolladores creen aplicaciones móviles conectadas mediante AWS. Este SDK también permite las comunicaciones de dispositivos MQTT y llamar a las API de la AWS IoT CoreServicios de . Para obtener más información, consulte los siguientes:

- [AWSSDK para móviles para Android en GitHub](#)
- [AWSArchivo léame de SDK para móviles de Android](#)
- [AWSEjemplos de SDK para móviles para Android](#)
- [AWSReferencia de la API de SDK for Android](#)
- [Documentación de referencia de la clase AWSocketClient](#)

iOS

AWS Mobile SDK for iOS

La AWS Mobile SDK for iOS es un kit de desarrollo de software de código abierto, distribuido bajo licencia de Apache Open Source. El SDK for iOS proporciona una biblioteca, muestras de código y documentación para ayudar a los desarrolladores a crear aplicaciones móviles conectadas mediante AWS. Este SDK también permite las comunicaciones de dispositivos MQTT y llamar a las API de la AWS IoT CoreServicios de . Para obtener más información, consulte los siguientes:

- [AWS Mobile SDK for iOS en GitHub](#)
- [AWSSDK for iOS Readme](#)
- [AWSSDK for iOS Samples](#)
- [AWS IoT Documentos de referencia de clase en el AWSSDK para iOS](#)

API de REST de AWS IoT CoreServicios de

Las API REST de AWS IoT Core se puede llamar directamente a los servicios mediante solicitudes HTTP.

- URL del punto de enlace

Los extremos de servicio que exponen las API REST de AWS IoT Core los servicios varían según la región y se enumeran en [AWS IoT Core Cuotas y puntos de enlace de](#). Debe utilizar el punto final de la región que tiene el AWS IoT recursos a los que desea obtener acceso, porque AWS IoT los recursos son específicos de cada región.

- Autenticación

Las API REST de AWS IoT Core usan servicios AWS Credenciales de IAM para autenticación. Para obtener más información, consulte [Firma de AWS Solicitudes API de](#) en la AWS Referencia general de .

- Referencia de la API

Para obtener información sobre las funciones específicas proporcionadas por las API REST de la AWS IoT Core servicios, consulte:

- [Referencia de API para IoT](#).
- [Referencia de API para datos de IoT](#).
- [Referencia de API para datos de trabajos de IoT](#).
- [Referencia de API para túneles seguros de IoT](#).

Conexión de dispositivos aAWS IoT

Conectarse dispositivos aAWS IoT y otros servicios a través de AWS IoT Core. A través de AWS IoT Core, los dispositivos envían y reciben mensajes mediante endpoints de dispositivo específicos de su cuenta. La sección llamada “[SDK de dispositivos de AWS IoT](#)” (p. 80) admite comunicaciones de dispositivos mediante los protocolos MQTT y WSS. Para obtener más información acerca de los protocolos que pueden usar los dispositivos, consulte la sección llamada “[Protocolos de comunicación de dispositivos](#)” (p. 81).

El agente de mensajes

AWS IoT administra la comunicación de dispositivos a través de un agente de mensajes. Los dispositivos y los clientes publican mensajes en el agente de mensajes y también se suscriben a los mensajes que publica el agente de mensajes. Los mensajes se identifican mediante una aplicación definida [Tema de \(p. 98\)](#). Cuando el agente de mensajes recibe un mensaje publicado por un dispositivo o cliente, vuelve a publicar ese mensaje en los dispositivos y clientes que se han suscrito al tema del mensaje. El agente de mensajes también reenvía mensajes a la AWS IoT [Reglas de \(p. 472\)](#) motor, que puede actuar sobre el contenido del mensaje.

AWS IoT seguridad de mensajes

Conexiones de dispositivos aAWS IoT usan la sección llamada “[Certificados de cliente X.509](#)” (p. 298) y AWS firma V4 para la autenticación. Las comunicaciones de dispositivos están protegidas por TLS versión 1.2 y AWS IoT requiere que los dispositivos envíen la extensión de indicación de nombre de servidor (SNI) cuando se conectan. Para obtener más información, consulte [Seguridad de transporte en AWS IoT](#).

AWS IoT datos de dispositivos y puntos finales de servicio

Cada cuenta tiene varios extremos de dispositivo que son exclusivos de la cuenta y admiten funciones específicas de IoT. La AWS IoT los endpoints de datos de dispositivos admiten un protocolo de publicación/suscripción diseñado para las necesidades de comunicación de los dispositivos IoT; sin embargo, otros clientes, como aplicaciones y servicios, también pueden utilizar esta interfaz si su aplicación requiere las funciones especializadas que proporcionan estos endpoints. La AWS IoT los endpoints de servicio de dispositivos admiten el acceso centrado en el dispositivo a los servicios de seguridad y administración.

Para conocer el punto final de datos del dispositivo de tu cuenta, puedes encontrarlo en el [Configuración de AWS IoT Core](#) consola de .

Para conocer el punto final del dispositivo de su cuenta para un fin específico, incluido el punto final de datos del dispositivo, utilice el comando `aws iot describe-endpoint` CLI que se muestra aquí, o la `DescribeEndpoint` API de REST y proporcione la `endpointType` valor de parámetro de la siguiente tabla.

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

Este comando devuelve un `iot-endpoint` en el siguiente formato: `account-specific-prefix.iot.aws-region.amazonaws.com`.

La `DescribeEndpoint` no es necesario consultar la API cada vez que se conecta un nuevo dispositivo. Los puntos finales que crea persisten para siempre y no cambian una vez creados.

Cada cliente tiene un `iot:Data-AT` y `iot>Data`. Cada punto de enlace utiliza un certificado X.509 para autenticar al cliente. Recomendamos a los clientes que utilicen el tipo de punto de enlace `iot:Data-AT` más reciente para evitar problemas relacionados con la desconfianza generalizada en las entidades.

de certificación de Symantec. Proporcionamos el `iot:Data` punto de enlace para dispositivos para recuperar datos de puntos de enlace antiguos que utilizan certificados VeriSign para compatibilidad con versiones anteriores. Para obtener más información, consulte [Autenticación del servidor](#).

AWS IoT endpoints para dispositivos

Propósito del punto de enlace	endpointType value	Descripción
AWS IoT Core- operaciones del plano de datos	<code>iot:Data-ATS</code>	<p>Se utiliza para enviar y recibir datos desde y hacia el agente de mensajes, Sombra de dispositivos (p. 627), y Motor de reglas de (p. 472) componentes de AWS IoT.</p> <p><code>iot:Data-ATS</code> devuelve un punto de enlace de datos firmado por ATS.</p>
AWS IoT Core- operaciones de plano de datos (heredado)	<code>iot:Data</code>	<code>iot:Data</code> devuelve un punto de enlace de datos firmados por VeriSign proporcionado para compatibilidad con versiones anteriores.
AWS IoT Core acceso a credenciales	<code>iot:CredentialProvider</code>	Se utiliza para intercambiar el certificado X.509 integrado de un dispositivo por credenciales temporales para conectarse directamente con otros AWS Servicios de . Para obtener más información acerca de cómo conectarse a otros servicios de AWS, consulte Autorizar llamadas directas a servicios de AWS (p. 375) .
AWS IoT Device Management- operaciones de datos de trabajos	<code>iot:Jobs</code>	Se utiliza para permitir que los dispositivos interactúen con el AWS IoT Servicio de Jobs utilizando la APIs HTTPS de dispositivos de trabajos (p. 746) .
AWS IoT Operaciones del Device Advisor	<code>iot:DeviceAdvisor</code>	Tipo de punto final de prueba utilizado para probar dispositivos con Device Advisor. Para obtener más información, consulte ??? (p. 1058).
AWS IoT Core data beta (vista previa)	<code>iot:Data-Beta</code>	Tipo de endpoint reservado para versiones beta. Para obtener información acerca de su uso actual, consulte ??? (p. 116).

También puede usar su propio nombre de dominio completo (FQDN), como `example.com` y el certificado de servidor asociado al que conectar dispositivos AWS IoT mediante el uso de [the section called "Puntos de enlace configurables" \(p. 116\)](#).

SDK de dispositivos de AWS IoT

La AWS IoT Los SDK de dispositivos te ayudan a conectar tus dispositivos IoT a AWS IoT. Los son compatibles con MQTT y MQTT a través de protocolos WSS.

La AWS IoT Los SDK de dispositivos difieren de los AWS SDK en que el AWS IoT Los SDK de dispositivos admiten las necesidades de comunicación especializadas de los dispositivos IoT, pero no admiten todos los servicios admitidos por el AWS SDK. La AWS IoT Los SDK de dispositivos son compatibles con el AWS SDK que admiten todos los AWS servicios; sin embargo, utilizan métodos de autenticación diferentes y se conectan a distintos endpoints, lo que podría hacer utilizando el AWS Los SDK no son prácticos en un dispositivo IoT.

Dispositivos móviles

La [sección titulada "AWS Mobile SDK" \(p. 76\)](#) admite comunicaciones de dispositivos MQTT, algunas de las AWS IoT API de servicio y las API de otros AWS servicios de . Si estás desarrollando en un dispositivo móvil compatible, revisa su SDK para ver si es la mejor opción para desarrollar tu solución IoT.

C++

AWS IoT C++ Device SDK

La AWS IoT C++ Device SDK permite a los desarrolladores compilar aplicaciones conectadas mediante AWSy las API del AWS IoT Core servicios de . En concreto, este SDK se diseñó para los dispositivos que no tienen limitación de recursos y requieren características avanzadas, como la puesta en cola de mensajes, la compatibilidad con varios procesos y las características de idioma más actualizadas. Para obtener más información, consulte los siguientes:

- [AWS IoT Device SDK C++ v2 en GitHub](#)
- [AWS IoT Archivo léame del SDK de dispositivos C++ v2](#)
- [AWS IoT Muestras del Device SDK C++ v2](#)
- [AWS IoT Documentación de la API de C++ v2 del SDK del dispositivo](#)

Python

AWS IoTDevice SDK for Python

La AWS IoTDevice SDK para Python permite a los desarrolladores escribir scripts de Python para tener acceso con sus dispositivos a la AWS IoT plataforma mediante MQTT o MQTT sobre protocolo WebSocket Secure (WSS). Conectando sus dispositivos a las API del AWS IoT Core servicios, los usuarios pueden trabajar de forma segura con el agente de mensajes, las reglas y el servicio Device Shadow que AWS IoT Core proporciona y con otros AWS servicios como AWS Lambda, Amazon Kinesis y Amazon S3 y más.

- [AWS IoT Device SDK for Python v2 en GitHub](#)
- [AWS IoT Device SDK for Python v2 Readme](#)
- [AWS IoT Ejemplos de Device SDK for Python v2](#)
- [AWS IoT Documentación de la API de Device SDK for Python v2](#)

JavaScript

AWS IoTDevice SDK for JavaScript

La AWS IoTDevice SDK para JavaScript permite a los desarrolladores escribir aplicaciones JavaScript que tengan acceso a las API de la AWS IoT Core utilizando MQTT o MQTT a través del protocolo

WebSocket. Se puede utilizar en entornos de Node.js y aplicaciones de navegador. Para obtener más información, consulte los siguientes:

- [AWS IoT Device SDK for JavaScript v2 en GitHub](#)
- [Archivo léame de AWS IoT Device SDK for JavaScript v2](#)
- [AWS IoT Ejemplos de Device SDK for JavaScript v2](#)
- [AWS IoT Documentación de la API de Device SDK for JavaScript v2](#)

Java

AWS IoTDevice SDK para Java

La AWS IoTDevice SDK for Java permite a los desarrolladores de Java tener acceso a las API de la AWS IoT Core mediante MQTT o MQTT mediante el protocolo WebSocket. El SDK admite el servicio Device Shadow. Puede tener acceso a las sombras mediante los métodos GET, UPDATE y DELETE de HTTP. El SDK es también compatible con un modelo de acceso de sombra simplificado, lo que permite a los desarrolladores intercambiar datos con sombras utilizando métodos getter y setter, sin tener que serializar o deserializar documentos JSON. Para obtener más información, consulte los siguientes:

- [AWS IoT Device SDK for Java v2 en GitHub](#)
- [Archivo léame de AWS IoT Device SDK for Java v2](#)
- [AWS IoT Ejemplos de Device SDK for Java v2](#)
- [AWS IoT Documentación de la API de Device SDK for Java v2](#)

Embedded C

AWS IoTDevice SDK for Embedded C

Important

Este SDK está diseñado para ser utilizado por desarrolladores de software integrado experimentados.

La AWS IoT Device SDK para Embedded C (C-SDK) es un conjunto de archivos de origen C bajo la licencia de código abierto MIT, que se puede utilizar en aplicaciones insertadas para establecer conexiones seguras de dispositivos IoT a AWS IoT Core. Incluye MQTT, analizador JSON y AWS IoT Librerías Device Shadow y otros. Se distribuye en forma de origen y está diseñado para integrarse en el firmware cliente junto con código de aplicación, otras bibliotecas y, opcionalmente, un RTOS (sistema operativo en tiempo real).

AWS IoT Device SDK para Embedded C generalmente se dirige a dispositivos con limitaciones de recursos que requieren un tiempo de ejecución optimizado del lenguaje C. Puede usar el SDK en cualquier sistema operativo y alojarlo en cualquier tipo de procesador (por ejemplo, MCU y MPU). Si su dispositivo tiene suficiente memoria y recursos de procesamiento disponibles, le recomendamos que utilice uno de los otros AWS IoT SDK para dispositivos y móviles, como la AWS IoTDevice SDK for C++, Java, JavaScript o Python.

Para obtener más información, consulte los siguientes:

- [AWS IoT Device SDK para Embedded C en GitHub](#)
- [AWS IoT Device SDK for Embedded C Readme](#)
- [AWS IoTDevice SDK for Embedded C Samples](#)

Protocolos de comunicación de dispositivos

AWS IoT Core es compatible con dispositivos y clientes que utilizan los protocolos MQTT y MQTT a través de WebSocket Secure (WSS) para publicar y suscribirse a mensajes, y dispositivos y clientes que utilizan el protocolo HTTPS para publicar mensajes. Todos los protocolos admiten IPv4 e IPv6. En esta sección se describen las distintas opciones de conexión para dispositivos y clientes.

TLS v1.2

AWS IoT Core utiliza [TLS versión 1.2](#) para cifrar todas las comunicaciones. Los clientes también deben enviar el [Extensión TLS de la indicación de nombre de servidor \(SNI\)](#). Se rechazan los intentos de conexión que no incluyen el SNI. Para obtener más información, consulte [Seguridad de transporte en AWS IoT](#).

La [SDK de dispositivos de AWS IoT \(p. 80\)](#) admite MQTT y MQTT a través de WSS y admite los requisitos de seguridad de las conexiones de clientes. Le recomendamos que utilice la [SDK de dispositivos de AWS IoT \(p. 80\)](#) para conectar clientes a AWS IoT.

Protocolos, mapeos de puertos y autenticación

La forma en que un dispositivo o cliente se conecta al agente de mensajes mediante un endpoint de dispositivo depende del protocolo que utilice. En la siguiente tabla, se enumeran los protocolos que la AWS IoT compatibilidad con endpoints de dispositivos y los métodos y puertos de autenticación que utilizan.

Protocolos, autenticación y mapeos de puerto

Protocolo	Operaciones admitidas	Autenticación	Puerto	Nombre del protocolo ALPN
MQTT a través de WebSocket	Publique, suscriba	Signature Version 4	443	N/A
MQTT a través de WebSocket	Publique, suscriba	Autenticación personalizada	443	N/A
MQTT	Publique, suscriba	Certificado de cliente X.509	443 [†]	x-amzn-mqtt-ca
MQTT	Publique, suscriba	Certificado de cliente X.509	8883	N/A
MQTT	Publique, suscriba	Autenticación personalizada	443 [†]	mqtt
HTTPS	Publicar solo	Signature Version 4	443	N/A
HTTPS	Publicar solo	Certificado de cliente X.509	443 [†]	x-amzn-http-ca
HTTPS	Publicar solo	Certificado de cliente X.509	8443	N/A
HTTPS	Publicar solo	Autenticación personalizada	443	N/A

Negociación de protocolo de capa de aplicación (ALPN)

[†]Los clientes que se conectan en el puerto 443 con autenticación de certificado de cliente X.509 deben implementar el [Negociación de protocolo de capa de aplicación \(ALPN\)](#) TLS y utilice la [Nombre del protocolo ALPN](#) listado en la lista de ProtocolNameList ALPN enviada por el cliente como parte del `clientHello` message.

En el puerto 443, elIoT: data-ATS (p. 79)endpoint admite ALPN x-amzn-http-ca HTTP, pero elIoT: empleos (p. 79)endpoint no lo hace.

En el puerto 8443 HTTPS y el puerto 443 MQTT con ALPN x-amzn-mqtt-ca,autenticación personalizada (p. 319)Las no se pueden usar.

Los clientes se conectan a susCuenta de AWSendpoints del dispositivo. Consulte[the section called “AWS IoTdatos de dispositivos y puntos finales de servicio” \(p. 78\)](#)para obtener información acerca de cómo buscar los puntos de enlace de dispositivo de su cuenta.

Note

AWSLos SDK no requieren la URL completa. Solo requieren el nombre de host del endpoint, como [elpubsub.pyEjemplo de paraAWS IoTDevice SDK for Python on GitHub](#). Al pasar toda la URL tal y como se indica en la tabla siguiente, se puede generar un error, como un nombre de host no válido.

Conexión a AWS IoT Core

Protocolo	URL o punto de enlace
MQTT	<code>iot-endpoint</code>
MQTT sobre WSS	<code>wss://iot-endpoint/mqtt</code>
HTTPS	<code>https://iot-endpoint/topics</code>

Selección de un protocolo para la comunicación de su dispositivo

Para la mayoría de las comunicaciones de dispositivos IoT a través de los endpoints del dispositivo, querrá utilizar MQTT o MQTT a través de protocolos WSS; sin embargo, los endpoints del dispositivo también admiten HTTPS. En la tabla siguiente se compara cómoAWS IoT Coreutiliza los dos protocolos para la comunicación del dispositivo.

AWS IoTprotocolos de dispositivo en paralelo

Característica	MQTT (p. 85)	HTTPS (p. 95)
Soporte de publicación/ suscripción	Publique y suscriba	Publicar solo
Compatibilidad con SDK	AWSSDKs de dispositivos (p. 80) admite protocolos MQTT y WSS	Sin compatibilidad con SDK, pero puede utilizar métodos específicos del idioma para realizar solicitudes HTTPS
Soporte de calidad de servicio	Niveles de QoS de MQTT 0 y 1 (p. 86)	QoS se admite mediante la transferencia de un parámetro de cadena de consulta? <code>qos=qos</code> donde el valor puede ser 0 o 1. Puede agregar esta cadena de consulta para publicar un mensaje con el valor QoS que desee.
Se pueden omitir mensajes recibidos mientras el dispositivo estaba sin conexión	Sí	No
clientIdssoporte de campo	Sí	No

Característica	MQTT (p. 85)	HTTPS (p. 95)
Detección de desconexión de dispositivos	Sí	No
Comunicaciones seguras	Sí. Consulte Protocolos, mapeos de puertos y autenticación (p. 82)	Sí. Consulte Protocolos, mapeos de puertos y autenticación (p. 82)
Definiciones de temas	Aplicación definida	Aplicación definida
Formato de los datos del mensaje	Aplicación definida	Aplicación definida
Sobrecarga del protocolo	Bajar	Más alto
Consumo de energía	Bajar	Más alto

Límites de duración de conexión

No se garantiza que las conexiones HTTPS duren más que el tiempo que tarda en recibir y responder a las solicitudes.

La duración de la conexión MQTT depende de la función de autenticación que utilice. En la siguiente tabla se muestra la duración máxima de la conexión en condiciones ideales para cada entidad.

Duración de la conexión MQTT por función de autenticación

Característica	Duración máxima *
Certificado de cliente X.509	1 a 2 semanas
Autenticación personalizada	1 a 2 semanas
Signature Version 4	Hasta 24 horas

* Las no garantizadas

Con los certificados X.509 y la autenticación personalizada, la duración de la conexión no tiene límite, pero puede durar tan solo unos minutos. Las interrupciones de la conexión pueden ocurrir por diversos motivos. En la siguiente lista se incluyen algunos de los motivos más comunes.

- Interrupciones de disponibilidad wifi
- Interrupciones de la conexión del proveedor de servicios de Internet (ISP)
- Parches de servicio
- Implementaciones de servicios
- Escalado automático de servicios
- Host de servicio no disponible
- Problemas y actualizaciones del balanceador de carga
- Errores del cliente

Los dispositivos deben implementar estrategias para detectar desconexiones y volver a conectarse. Para obtener información acerca de los eventos de desconexión y la guía sobre cómo gestionarlos, consulte [??? \(p. 1111\)](#) en [??? \(p. 1111\)](#).

MQTT

MQTT es un protocolo de mensajería ligero y ampliamente adoptado que está diseñado para dispositivos limitados. AWS IoT soporta MQTT se basa en el [Especificación MQTT v3.1.1](#), con algunas diferencias. Para obtener más información sobre cómo AWS IoT difiere de la especificación MQTT v3.1.1, consulte [the section called "AWS IoT diferencias con respecto a la especificación MQTT versión 3.1.1" \(p. 95\)](#).

AWS IoT Core admite conexiones de dispositivos que utilizan el protocolo MQTT y el protocolo MQTT sobre WSS y que se identifican mediante un ID de cliente. La [SDK de dispositivos de AWS IoT \(p. 80\)](#) admite ambos protocolos y son las formas recomendadas de conectar dispositivos a AWS IoT. La [AWS IoT Los SDK de dispositivos](#) admiten las funciones necesarias para que los dispositivos y los clientes se conecten y accedan a los servicios de AWS IoT. Los SDK de dispositivos admiten los protocolos de autenticación que los servicios requieren y los requisitos de ID de conexión que requieren el protocolo MQTT y los protocolos MQTT sobre WSS. Para obtener información acerca de cómo conectarse a AWS IoT utilizando el [AWS SDK de dispositivos y enlaces a ejemplos de AWS IoT](#) en los idiomas admitidos, consulte [the section called "Conexión con MQTT mediante el AWS IoT SDKs de dispositivos" \(p. 85\)](#). Para obtener más información acerca de los métodos de autenticación y las asignaciones de puertos para mensajes MQTT, consulte [Protocolos, mapeos de puertos y autenticación \(p. 82\)](#).

Si bien recomendamos utilizar el [AWS IoT SDK de dispositivos](#) a los que conectarse a AWS IoT, no son obligatorias. Si no usa el [AWS IoT](#) sin embargo, los SDK de dispositivos deben proporcionar la conexión y la seguridad de comunicación necesarias. Los clientes deben enviar el [Extensión TLS de la indicación de nombre de servidor \(SNI\)](#) en la solicitud de conexión. Se rechazan los intentos de conexión que no incluyen el SNI. Para obtener más información, consulte [Seguridad de transporte en AWS IoT](#). Clientes que utilizan usuarios de IAM y AWS las credenciales para autenticar clientes deben proporcionar el correcto [Signature Version 4](#) autenticación.

Conexión con MQTT mediante el AWS IoT SDKs de dispositivos

Esta sección contiene enlaces a la [AWS IoT SDK de dispositivo](#) y el código fuente de los programas de ejemplo que ilustran cómo conectar un dispositivo a AWS IoT. Las aplicaciones de ejemplo enlazadas aquí muestran cómo conectarse a AWS IoT utilizando el protocolo MQTT y MQTT sobre WSS.

C++

Uso de AWS IoT C++ Device SDK para conectar dispositivos

- [Código fuente de una aplicación de ejemplo que muestra un ejemplo de conexión MQTT en C++](#)
- [AWS IoT C++ Device SDK v2 en GitHub](#)

Python

Uso de AWS IoT Device SDK for Python para conectar dispositivos

- [Código fuente de una aplicación de ejemplo que muestra un ejemplo de conexión MQTT en Python](#)
- [AWS IoT Device SDK for Python v2 en GitHub](#)

JavaScript

Uso de AWS IoT Device SDK for JavaScript para conectar dispositivos

- [Código fuente de una aplicación de ejemplo que muestra un ejemplo de conexión MQTT en JavaScript](#)
- [AWS IoT Device SDK for JavaScript v2 en GitHub](#)

Java

Uso de AWS IoTDevice SDK for Java para conectar dispositivos

- [Código fuente de una aplicación de ejemplo que muestra un ejemplo de conexión MQTT en Java](#)
- [AWS IoT Device SDK for Java v2 en GitHub](#)

Embedded C

Uso de AWS IoTDevice SDK for Embedded C para conectar dispositivos

Important

Este SDK está diseñado para ser utilizado por desarrolladores de software integrado experimentados.

- [Código fuente de una aplicación de ejemplo que muestra un ejemplo de conexión MQTT en C embebido](#)
- [AWS IoT Device SDK para Embedded C en GitHub](#)

Opciones de calidad de servicio (QoS) de MQTT

AWS IoT y la AWS IoT Los SDKs de dispositivos admiten la [Niveles de calidad de servicio \(QoS\) de MQTT](#) 0 y 1. El protocolo MQTT define un tercer nivel de QoS, nivel 2, pero AWS IoT no lo admite. Solo el protocolo MQTT admite la función QoS. HTTPS admite QoS al pasar un parámetro de cadena de consulta?qos=qos donde el valor puede ser 0 o 1.

En esta tabla se describe cómo afecta cada nivel de QoS a los mensajes publicados en y por el agente de mensajes.

Con un nivel de QoS de...	El mensaje es...	Comentarios
QoS nivel 0	Enviado cero o más veces	Este nivel debe utilizarse para mensajes que se envían a través de enlaces de comunicación fiables o que se pueden perder sin problemas.
QoS de nivel 1	Enviado al menos una vez, y luego repetidamente hasta que <code>PUBLISH</code> se recibe la respuesta	El mensaje no se considera completo hasta que el remitente recibe un <code>PUBLISH</code> respuesta para indicar la entrega satisfactoria.

Uso de sesiones persistentes de MQTT

Las sesiones persistentes almacenan las suscripciones y los mensajes de un cliente, con una calidad de servicio (QoS) de 1, que el cliente no ha reconocido. Cuando un dispositivo desconectado se vuelve a conectar a una sesión persistente, la sesión se reanuda, se restablecen sus suscripciones y los mensajes suscritos recibidos antes de la reconexión y que el cliente no ha confirmado se envían al cliente.

Creación de una sesión persistente

Puede crear una sesión persistente de MQTT enviando una `CONNECT` mensaje y configuración `cleanSession` marcar para 0. Si no existe ninguna sesión para el cliente que envía la `CONNECT`, se crea una sesión persistente nueva. Si ya existe una sesión para el cliente, el cliente reanuda la sesión existente.

Operaciones durante una sesión persistente

Los clientes utilizan la `sessionPresent` atributo en la conexión confirmada (CONNACK) para determinar si existe una sesión persistente. Si `sessionPresent` es 1, existe una sesión persistente y todos los mensajes almacenados para el cliente se entregan al cliente inmediatamente después de que el cliente recibe el CONNACK, según se explica en [Tráfico de mensajes tras la reconexión a una sesión persistente \(p. 87\)](#). Si `sessionPresent` es 0, no es necesario que el cliente vuelva a suscribirse. Sin embargo, si `sessionPresent` es 0, no existe ninguna sesión persistente y que el cliente debe volver a suscribirse a sus filtros de temas.

Una vez que el cliente se une a una sesión persistente, puede publicar mensajes y suscribirse a filtros de temas sin ningún indicador adicional en cada operación.

Tráfico de mensajes tras la reconexión a una sesión persistente

Una sesión persistente representa una conexión continua entre un cliente y un agente de mensajes de MQTT. Cuando un cliente se conecta al intermediario de mensajes de mediante una sesión persistente, el agente de mensajes guarda todas las suscripciones que el cliente realiza durante la conexión. Cuando el cliente se desconecta, el agente de mensajes almacena los mensajes QoS 1 sin confirmar y los mensajes QoS 1 nuevos publicados en los temas a los que el cliente está suscrito. Los mensajes se almacenan según el límite de la cuenta, los mensajes que superen ese límite se eliminarán. Para obtener más información acerca de los límites de mensajes persistentes, consulte [AWS IoT Core Cuotas y puntos de enlace de](#). Cuando el cliente vuelve a conectarse su sesión persistente, todas las suscripciones se restablecen y todos los mensajes almacenados se envían al cliente a una velocidad máxima de 10 mensajes por segundo.

Tras la reconexión, los mensajes almacenados se envían al cliente, a una velocidad limitada a 10 mensajes almacenados por segundo, junto con cualquier tráfico de mensajes actual hasta que el `Publish requests per second per connection` se alcanza el límite. Dado que la velocidad de entrega de los mensajes almacenados es limitada, tardará varios segundos en entregar todos los mensajes almacenados si una sesión tiene más de 10 mensajes almacenados para entregarlos tras la reconexión.

Finalización de una sesión persistente

Las condiciones siguientes describen cómo pueden terminar las sesiones persistentes.

- Cuando transcurre el tiempo de caducidad de la sesión persistente. El temporizador de caducidad de la sesión persistente se inicia cuando el agente de mensajes detecta que un cliente se ha desconectado, ya sea por la desconexión del cliente o por el tiempo de espera de la conexión.
- Cuando el cliente envía un `UNCONNECT` mensaje que establece el `cleanSession` marcar para 1.

Note

Los mensajes almacenados que esperan ser enviados al cliente cuando finaliza una sesión se descartan; sin embargo, se siguen facturando a la velocidad de mensajería estándar, aunque no se hayan podido enviar. Para obtener más información acerca de los precios de los mensajes, consulte [AWS IoT Core Precios](#). Puede configurar el intervalo de tiempo de caducidad.

Reconexión después de que haya caducado una sesión persistente

Si un cliente no se vuelve a conectar a su sesión persistente antes de que caduque, la sesión finaliza y se descartan los mensajes almacenados. Cuando un cliente se vuelve a conectar después de que la sesión haya caducado con `uncleanSession` marcar para 0, el servicio crea una nueva sesión persistente. Las suscripciones o mensajes de la sesión anterior no están disponibles para esta sesión porque se descartaron cuando caducó la sesión anterior.

Cargos por mensajes de sesión persistentes

Los mensajes se cargan a tu cuenta de AWS cuando el agente de mensajes envía un mensaje a un cliente o a una sesión persistente sin conexión. Cuando un dispositivo sin conexión con una sesión persistente se vuelve a conectar y reanuda su sesión, los mensajes almacenados se entregan en el dispositivo y se cargan de nuevo en su cuenta. Para obtener más información acerca de los precios de los mensajes, consulte [AWS IoT Core price - Mensajería](#).

El tiempo de caducidad de la sesión persistente por defecto de una hora puede aumentarse utilizando el proceso de aumento del límite estándar. Ten en cuenta que aumentar el tiempo de caducidad de la sesión podría aumentar los cargos por mensaje porque el tiempo adicional podría permitir que se almacenen más mensajes para el dispositivo sin conexión y esos mensajes adicionales se cargarían a tu cuenta con la tasa de mensajería estándar. El tiempo de caducidad de la sesión es aproximado y una sesión podría persistir hasta 30 minutos más que el límite de la cuenta; sin embargo, una sesión no será inferior al límite de la cuenta. Para obtener más información acerca de los límites de sesión, consulte [AWS Quotas de servicio](#).

Uso de mensajes conservados de MQTT

AWS IoT Core admite el indicador RETAIN descrito en [MQTT v3.1.1](#). Cuando un cliente establece el indicador RETAIN en un mensaje MQTT que publica, AWS IoT Core guarda el mensaje. A continuación, se puede enviar a los nuevos suscriptores, recuperarse llamando al [GetRetainedMessage](#) y se visualiza en la [AWS IoT consola](#). AWS IoT Core almacena los mensajes conservados durante tres años después de la última vez que se actualizaron o se accedió a ellos. Después de tres años, los mensajes se eliminan.

Ejemplos de uso de mensajes retenidos de MQTT

- Como mensaje de configuración inicial

Los mensajes retenidos de MQTT se envían a un cliente después de que el cliente se suscribe a un tema. Si desea que todos los clientes que se suscriban a un tema reciban el mensaje conservado de MQTT inmediatamente después de su suscripción, puede publicar un mensaje de configuración con el indicador RETAIN establecido. Los clientes suscritos también reciben actualizaciones de esa configuración cada vez que se publica un nuevo mensaje de configuración.

- Como último mensaje de estado conocido

Los dispositivos pueden establecer el indicador RETAIN en los mensajes de estado actual para que AWS IoT Core los guarde. Cuando las aplicaciones se conectan o se vuelven a conectar, pueden suscribirse a este tema y obtener el último estado notificado inmediatamente después de suscribirse al tema del mensaje conservado. De esta forma, pueden evitar tener que esperar hasta el siguiente mensaje del dispositivo para ver el estado actual.

En esta sección:

- [Tareas comunes con mensajes conservados de MQTT en AWS IoT Core \(p. 88\)](#)
- [Mensajes de facturación y retenidos \(p. 90\)](#)
- [Comparación de mensajes retenidos de MQTT y sesiones persistentes de MQTT \(p. 91\)](#)
- [MQTT conservó los mensajes y AWS IoT Sombras del dispositivo \(p. 93\)](#)

Tareas comunes con mensajes conservados de MQTT en AWS IoT Core

AWS IoT Core guarda los mensajes MQTT con la marca RETAIN establecida. Estos mensajes retenidos se envían a todos los clientes que se han suscrito al tema, como mensaje MQTT normal, y también se almacenan para enviarlos a nuevos suscriptores del tema.

Los mensajes retenidos de MQTT requieren acciones políticas específicas para autorizar a los clientes a acceder a ellos. Para obtener ejemplos de uso de políticas de mensajes conservados, consulte [Ejemplos de políticas de mensajes conservados \(p. 366\)](#).

En esta sección se describen las operaciones comunes que implican mensajes retenidos.

- Creación de un mensaje conservado

El cliente determina si un mensaje se conserva cuando publica un mensaje MQTT. Los clientes pueden establecer el indicador RETAIN cuando publican un mensaje mediante unSDK de dispositivos (p. 1274). Las aplicaciones y los servicios pueden establecer el indicador RETAIN cuando utilizan elPublishacción para publicar un mensaje MQTT.

Solo se conserva un mensaje por nombre de tema. Un nuevo mensaje con el conjunto de marcas RETAIN publicado en un tema reemplaza cualquier mensaje conservado que se haya enviado al tema anteriormente.

NOTA: No puede publicar en unTema reservado (p. 100)con el conjunto de banderas RETAIN.

- Suscripción a un tema de mensaje conservado

Los clientes se suscriben a temas de mensajes conservados como lo harían con cualquier otro tema de mensaje MQTT. Los mensajes retenidos recibidos al suscribirse a un tema de mensaje conservado tienen la marca RETAIN establecida.

Los mensajes retenidos se eliminan deAWS IoT Corecuando un cliente publica un mensaje retenido con una carga útil de mensajes de 0 bytes en el tema del mensaje conservado. Los clientes que se hayan suscrito al tema del mensaje conservado también recibirán el mensaje de 0 bytes.

Al suscribirse a un filtro de temas comodín que incluye un tema de mensaje retenido, el cliente puede recibir mensajes posteriores publicados en el tema del mensaje retenido, pero no entrega el mensaje retenido tras la suscripción.

NOTA: Para recibir un mensaje conservado tras la suscripción, el filtro de temas de la solicitud de suscripción debe coincidir exactamente con el tema del mensaje conservado.

Los mensajes retenidos recibidos al suscribirse a un tema de mensaje conservado tienen el indicador RETAIN establecido. Los mensajes conservados que recibe un cliente suscrito después de la suscripción, no lo hacen.

- Recuperación de un mensaje conservado

Los mensajes retenidos se entregan a los clientes automáticamente cuando se suscriben al tema con el mensaje conservado. Para que un cliente reciba el mensaje conservado tras la suscripción, debe suscribirse al nombre exacto del tema del mensaje conservado. La suscripción a un filtro de temas comodín que incluye un tema de mensaje retenido permite al cliente recibir mensajes posteriores publicados en el tema del mensaje retenido, pero no entrega el mensaje retenido tras la suscripción.

Los servicios y las aplicaciones pueden listar y recuperar mensajes retenidos llamandoListRetainedMessagesyGetRetainedMessage.

No se impide a un cliente publicar mensajes en un tema de mensaje retenido sin establecer el indicador RETAIN. Esto podría provocar resultados inesperados, como el mensaje conservado no coincide con el mensaje recibido al suscribirse al tema.

- Listar temas de mensajes conservados

Puede listar los mensajes retenidos llamandoListRetainedMessagesy los mensajes conservados se pueden ver en laAWS IoTconsola.

- Obtener detalles de mensajes conservados

Puede obtener los detalles del mensaje conservado llamandoGetRetainedMessagey se pueden ver en laAWS IoTconsola.

- Conservación de un mensaje Will

MQTTVoluntadmessagesque se crean cuando se conecta un dispositivo se puede conservar configurando lawill RetainMarcadores deConnect Flag bits.

- Eliminar un mensaje conservado

Los dispositivos, aplicaciones y servicios pueden eliminar un mensaje retenido publicando un mensaje con el indicador RETAIN establecido y una carga útil de mensaje vacía (0 bytes) en el nombre del tema del mensaje retenido que eliminar. Dichos mensajes eliminan el mensaje conservado de AWS IoT Core, se envían a clientes con una suscripción al tema, pero no se conservan AWS IoT Core.

Los mensajes retenidos también se pueden eliminar de forma interactiva accediendo al mensaje conservado en la [AWS IoTconsola](#). Mensajes retenidos que se eliminan mediante la [AWS IoTconsola](#) también envía un mensaje de 0 bytes a los clientes que se han suscrito al tema del mensaje conservado.

Los mensajes retenidos no se pueden restaurar después de eliminarlos. Un cliente tendría que publicar un nuevo mensaje conservado para sustituir al mensaje eliminado.

- Depuración y solución de problemas de mensajes retenidos

La [AWS IoTconsola](#) proporciona varias herramientas que le ayudarán a solucionar problemas de mensajes retenidos:

- La [Mensajes retenidos](#) página

La [Mensajes retenidos](#) (Se ha creado la [AWS IoTconsole](#)) proporciona una lista paginada de los mensajes conservados que su Cuenta ha almacenado en la región actual. Desde esta página puede hacer lo siguiente:

- Consulte los detalles de cada mensaje conservado, como la carga útil del mensaje, la QoS, la hora en que se recibió.
- Actualiza el contenido de un mensaje conservado.
- Eliminar un mensaje conservado.

- La [Cliente de prueba MQTT](#)

La [Cliente de prueba MQTT](#) (Se ha creado la [AWS IoTconsole](#)) puede suscribirse y publicar en temas de MQTT. La opción de publicación le permite establecer el indicador RETAIN en los mensajes que publica para simular cómo se comportan los dispositivos.

Algunos resultados inesperados podrían ser el resultado de estos aspectos de cómo se implementan los mensajes retenidos en AWS IoT Core.

- Límites de mensajes retenidos

Cuando una cuenta ha almacenado el número máximo de mensajes retenidos, AWS IoT Core devuelve una respuesta restringida a los mensajes publicados con el conjunto RETAIN y cargas útiles superiores a 0 bytes hasta que se eliminan algunos mensajes retenidos y el recuento de mensajes retenidos quede por debajo del límite.

- Orden de entrega de mensajes conservados

La secuencia de mensajes retenidos y de entrega de mensajes suscritos no está garantizada.

Mensajes de facturación y retenidos

Publicar mensajes con el indicador RETAIN establecido desde un cliente, mediante [AWS IoTconsola](#), o llamando [Publish](#) incurre en cargos de mensajería adicionales descritos en [AWS IoT Core price - Mensajería](#).

Recuperación de mensajes retenidos por un cliente, mediante [AWS IoTconsola](#), o llamando [GetRetainedMessage](#) incurre en cargos por mensajería además de los cargos por uso normal de la API. Los cargos adicionales se describen en [AWS IoT Core price - Mensajería](#).

MQTT [Voluntad messages](#) que se publican cuando un dispositivo se desconecta incurre de forma inesperada en cargos de mensajería descritos en [AWS IoT Core price - Mensajería](#).

Para obtener más información acerca de los costos de mensajería, consulte [AWS IoT Core price - Mensajería](#).

Comparación de mensajes retenidos de MQTT y sesiones persistentes de MQTT

Los mensajes retenidos y las sesiones persistentes son características estándar de MQTT 3.1.1 que permiten que los dispositivos reciban mensajes publicados mientras estaban sin conexión. Los mensajes retenidos se pueden publicar desde sesiones persistentes. En esta sección se describen aspectos clave de estas características y cómo funcionan entre sí.

	Mensajes retenidos	Sesiones persistentes	Mensajes conservados en sesiones persistentes
Características principales	<p>Los mensajes retenidos se pueden utilizar para configurar o notificar a grandes grupos de dispositivos después de conectarse.</p> <p>Los mensajes conservados también se pueden utilizar cuando desea que los dispositivos reciban solo el último mensaje publicado en un tema tras una reconexión.</p>	<p>Las sesiones persistentes son útiles para dispositivos que tienen conectividad intermitente y pueden perder varios mensajes importantes.</p> <p>Los dispositivos se pueden conectar con una sesión persistente para recibir mensajes enviados mientras están sin conexión.</p>	Los mensajes retenidos se pueden utilizar tanto en sesiones regulares como persistentes.
Ejemplos	Los mensajes retenidos pueden proporcionar información sobre la configuración de los dispositivos sobre su entorno cuando se conectan. La configuración inicial podría incluir una lista de otros temas de mensajes a los que debe suscribirse o información sobre cómo debe configurar su zona horaria local.	Los dispositivos que se conectan a través de una red celular con conectividad intermitente podrían utilizar sesiones persistentes para evitar que se pierdan mensajes importantes que se envían mientras un dispositivo está fuera de la cobertura de la red o necesita apagar su radio celular.	El dispositivo celular de la muestra de sesión persistente podría utilizar un mensaje retenido para recibir su configuración inicial en su conexión inicial.
Mensajes recibidos en la suscripción inicial a un tema	Tras suscribirse a un tema con un mensaje conservado, se recibe el mensaje conservado más reciente.	Tras suscribirse a un tema sin un mensaje retenido, no se recibe ningún mensaje hasta que se publique uno en el tema.	Tras suscribirse a un tema con un mensaje conservado, se recibe el mensaje conservado más reciente.
Temas suscritos tras la reconexión	Sin una sesión persistente, el cliente debe suscribirse a los temas tras la reconexión.	Los temas suscritos se restauran tras la reconexión.	Los temas suscritos se restauran tras la reconexión.

	Mensajes retenidos	Sesiones persistentes	Mensajes conservados en sesiones persistentes
Mensajes recibidos tras la reconexión	Tras suscribirse a un tema con un mensaje conservado, se recibe el mensaje conservado más reciente.	Todos los mensajes publicados con QOS = 1 y suscritos con QOS =1 mientras se desconectó el dispositivo se envían después de que el dispositivo se vuelva a conectar.	<p>Todos los mensajes publicados con QOS = 1 y suscritos con QOS =1 que se enviaron mientras se desconectó el dispositivo se envían después de volver a conectarse el dispositivo.</p> <p>Los mensajes retenidos actualizados de los temas a los que se suscribió el cliente también se envían al cliente.</p> <p>Si se publicó más de un mensaje retenido en un tema mientras el cliente estaba sin conexión, puede recibir varios mensajes conservados almacenados en ese tema después de volver a conectarse.</p>

	Mensajes retenidos	Sesiones persistentes	Mensajes conservados en sesiones persistentes
Caducidad de datos/sesión	<p>Los mensajes retenidos no caducan. Se almacenan hasta que se sustituyen o eliminan.</p> <p>Para obtener más información acerca de la caducidad de la sesión con sesiones persistentes, consulte the section called “Uso de sesiones persistentes de MQTT” (p. 86).</p>	<p>Las sesiones persistentes caducan si el cliente no se vuelve a conectar dentro del período de tiempo de espera. Una vez que caduca una sesión persistente, se eliminan las suscripciones del cliente y los mensajes guardados que se publicaron con QOS = 1 y a los que se suscribieron con QOS =1 mientras se desconectó el dispositivo.</p> <p>Para obtener más información acerca de la caducidad de la sesión con sesiones persistentes, consulte the section called “Uso de sesiones persistentes de MQTT” (p. 86).</p>	<p>Los mensajes retenidos no caducan. Se almacenan hasta que se reemplazan o eliminan incluso si se envían desde una sesión persistente que ha caducado. Una vez que caduca una sesión persistente, se eliminan las suscripciones del cliente y los mensajes guardados que se publicaron con QOS = 1 y a los que se suscribieron con QOS =1 mientras se desconectó el dispositivo.</p>

Para obtener información acerca de sesiones persistentes, consulte [the section called “Uso de sesiones persistentes de MQTT” \(p. 86\)](#).

Con Mensajes retenidos, el cliente de publicación determina si un mensaje debe conservarse y entregarse a un dispositivo después de conectarse, ya sea que haya tenido una sesión anterior o no. La elección de almacenar un mensaje la realiza el editor y el mensaje almacenado se entrega a todos los clientes actuales y futuros que se suscriban con suscripciones QoS 0 o QoS 1. Los mensajes retenidos guardan solo un mensaje sobre un tema determinado a la vez.

Cuando una cuenta ha almacenado el número máximo de mensajes retenidos, AWS IoT Core devuelve una respuesta restringida a los mensajes publicados con el conjunto RETAIN y cargas útiles superiores a 0 bytes hasta que se eliminan algunos mensajes retenidos y el recuento de mensajes retenidos quede por debajo del límite.

MQTT conservó los mensajes y AWS IoT Sombras del dispositivo

Los mensajes retenidos y las sombras de dispositivos conservan los datos de un dispositivo, pero se comportan de forma diferente y tienen diferentes propósitos. En esta sección se describen sus similitudes y diferencias.

	Mensajes retenidos	Sombras del dispositivo
La carga útil de mensajes tiene una estructura o esquema predefinidos	Según lo definido por la implementación. MQTT no especifica ninguna estructura o	AWS IoT admite una estructura de datos específica.

	Mensajes retenidos	Sombras del dispositivo
	esquema para su carga útil de mensajes.	
La actualización de la carga útil de mensajes genera mensajes de eventos	La publicación de un mensaje retenido envía el mensaje a los clientes suscritos, pero no genera mensajes de actualización adicionales.	La actualización de Device Shadow produce mensajes de actualización que describen el cambio .
Las actualizaciones de mensajes están numeradas	Los mensajes retenidos no se numeran automáticamente.	Los documentos Device Shadow tienen números de versión y marcas de hora automáticas.
La carga útil de un mensaje se adjunta a un recurso de cosa	Los mensajes retenidos no se adjuntan a un recurso de cosa.	Las sombras de dispositivos se adjuntan a un recurso de cosa.
Actualización de elementos individuales de la carga útil de mensajes	Los elementos individuales del mensaje no se pueden cambiar sin actualizar toda la carga útil del mensaje.	Los elementos individuales de un documento Device Shadow se pueden actualizar sin necesidad de actualizar todo el documento Device Shadow.
El cliente recibe datos de mensajes tras la suscripción	El cliente recibe automáticamente un mensaje retenido después de suscribirse a un tema con un mensaje retenido.	Los clientes pueden suscribirse a las actualizaciones de Device Shadow, pero deben solicitar el estado actual deliberadamente.
Indexación y capacidad de búsqueda	Los mensajes retenidos no se indexan para la búsqueda.	La indexación de flotas indexa los datos de Device Shadow para búsqueda y agregación.

Uso de ConnectAttributes

`ConnectAttributes` le permiten especificar qué atributos desea utilizar en el mensaje de conexión en sus políticas de IAM, tales como `PersistentConnect` y `LastWill`. `conConnectAttributes`, puede crear políticas que no permitan a los dispositivos acceso a nuevas funciones de forma predeterminada, lo que puede resultar útil si un dispositivo se ve comprometido.

`connectAttributes` es compatible con las siguientes características:

`PersistentConnect`

Usar `PersistentConnect` para guardar todas las suscripciones que el cliente realiza durante la conexión cuando la conexión entre el cliente y el intermediario se interrumpe.

`LastWill`

Usar `LastWill` para publicar un mensaje en la `LastWillTopic` cuando un cliente se desconecta inesperadamente.

De forma predeterminada, la política tiene una conexión no persistente y no se pasan atributos para esta conexión. Debe especificar una conexión persistente en su política de IAM si desea tener una.

Para `ConnectAttributes` ejemplos, consulte [Ejemplos de política de conexión \(p. 344\)](#).

AWS IoT diferencias con respecto a la especificación MQTT versión 3.1.1

La implementación del agente de mensajes se basa en el [Especificación MQTT v3.1.1](#), pero difiere de la especificación como se indica a continuación:

- AWS IoT solo permite la calidad de servicio (QoS) de MQTT únicamente los niveles 0 y 1. AWS IoT no admite la publicación ni la suscripción con QoS nivel 2. Cuando se solicita QoS nivel 2, el agente de mensajes de no envía PUBACK o SUBACK.
- En AWS IoT, suscribirse a un tema con QoS nivel 0 significa que un mensaje se entrega cero o más veces. Es posible que un mensaje se entregue más de una vez. Los mensajes que se entreguen más de una vez pueden enviarse con otro ID de paquete. En tal caso, la marca DUP no se establece.
- Cuando responde a una solicitud de conexión, el agente de mensajes envía un mensaje CONNACK. Este mensaje contiene una marca para indicar si la conexión reanuda una sesión anterior.
- Antes de enviar paquetes de control adicionales o una solicitud de desconexión, el cliente debe esperar a que se reciba el mensaje CONNACK en su dispositivo desde el [AWS IoT Agente de mensajes](#).
- Cuando un cliente se suscribe a un tema, puede haber un retraso entre el momento en que el agente de mensajes envía un SUBACK y el momento en que el cliente empieza a recibir nuevos mensajes coincidentes.
- Cuando un cliente utiliza el carácter comodín # en el filtro de temas para suscribirse a un tema, todas las cadenas en y por debajo de su nivel en la jerarquía de temas se corresponden. Sin embargo, el tema principal no coincide. Por ejemplo, una suscripción al tema `sensor/#` recibe mensajes publicados en los temas `sensor/sensor/temperature`, `sensor/temperature/sensor/room1`, pero no mensajes publicados en `sensor`. Para obtener más información acerca de los comodines, consulte [Filtros de temas \(p. 99\)](#).
- El agente de mensajes utiliza el ID de cliente para identificar a cada cliente. El ID de cliente se transfiere desde el cliente al agente de mensajes como parte de la carga de MQTT. Dos clientes con el mismo ID de cliente no pueden conectarse simultáneamente al agente de mensajes. Cuando un cliente se conecta al agente de mensajes mediante un ID de cliente que otro cliente está utilizando, se acepta la nueva conexión de cliente y se desconecta el cliente previamente conectado.
- En raras ocasiones, el agente de mensajes reenviará el mismo mensaje PUBLISH lógico con otro ID de paquete.
- La suscripción a filtros de temas que contienen un carácter comodín no puede recibir mensajes retenidos. Para recibir un mensaje conservado, la solicitud de suscripción debe contener un filtro de temas que coincida exactamente con el tema del mensaje conservado.
- El agente de mensajes de no garantiza el orden en que se reciben los mensajes y ACK.

HTTPS

Los clientes pueden publicar mensajes realizando solicitudes a la API REST utilizando los protocolos HTTP 1.0 o 1.1. Para obtener información sobre la autenticación y los mapeos de puertos utilizados por las solicitudes HTTP, consulte [Protocolos, mapeos de puertos y autenticación \(p. 82\)](#).

Note

A diferencia de MQTT, HTTPS no admite un valor `clientId`. Así que, mientras que `unclientId` está disponible cuando se utiliza MQTT, no está disponible cuando se utiliza HTTPS.

URL de mensajes HTTPS

Los dispositivos y clientes publican sus mensajes realizando solicitudes POST a un punto de enlace específico del cliente y a una dirección URL específica del tema:

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

- *IOT_data_endpoints* es el AWS IoT punto de enlace de datos de dispositivo (p. 78). Puede encontrar el punto de enlace en la AWS IoT en la página de detalles de la cosa o en el cliente mediante la AWS CLI comando:

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

El punto de enlace debería tener un aspecto similar al siguiente: a3qjEXAMPLEffp-ats.iot.us-west-2.amazonaws.com

- *url_encoded_topic_name* es el nombre del tema (p. 98) completo del mensaje enviado.

Ejemplos de códigos de mensajes HTTPS

Estos son algunos ejemplos de cómo enviar un mensaje HTTPS a AWS 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

Puede usar [curl](#) desde un cliente o dispositivo para enviar un mensaje a AWS IoT.

Para usar curl para enviar un mensaje desde un dispositivo cliente de AWS IoT

1. Consulte elcurlVersión de .

- a. En su cliente, ejecute este comando en un símbolo del sistema.

```
curl --help
```

En el texto de ayuda, busque las opciones TLS. Debería ver la opción `--tlsv1.2`.

- b. Si ve la opción `--tlsv1.2`, continúe.

- c. Si no ve el `--tlsv1.2` obtienes una command not founderror, es posible que tengas que actualizar o instalar curl en tu cliente o instalaropensslantes de continuar.

2. Instale los certificados en su cliente.

Copie los archivos de certificado que creó cuando registró su cliente (objeto) en la consola de AWS IoT. Asegúrese de que tiene estos tres archivos de certificado en su cliente antes de continuar.

- El archivo de certificado de CA (`Amazon-root-CA-1.pem`en este ejemplo).
- El archivo de certificado del cliente (`dispositivo.pem.crt`en este ejemplo).
- El archivo de clave privada del cliente (`private.pem.key`en este ejemplo).

3. Cree elcurllínea de comandos, sustituyendo los valores reemplazables para los de su cuenta y 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).

`--cacertAmazon-root-CA-1.pem`

El nombre de archivo y la ruta de acceso, si es necesario, del certificado de entidad de certificación para verificar el par.

`--certdispositivo.pem.crt`

El nombre y la ruta de acceso del archivo de certificado del cliente, si es necesario.

`--keyprivate.pem.key`

El nombre y la ruta del archivo de la clave privada del cliente, si es necesario.

`--request POST`

El tipo de solicitud HTTP (en este caso, POST).

`--data{ "mensaje": "Hola, mundo" }`

Los datos de HTTP POST que quiere publicar. En este caso, es una cadena JSON, con las comillas internas con el carácter de escape de barra invertida (\).

«https://iot-data-endpoint:8443/Temas/Tema_de? qos=1»

La URL de tu cliente AWS IoT extremo de datos del dispositivo, seguido del puerto HTTPS, :8443, seguida de la palabra clave, /topics/ y el nombre del tema, topic, en este caso. Especifique la calidad de servicio como parámetro de consulta, ?qos=1.

4. Abra el cliente de prueba MQTT en la AWS IoT consola de .

Siga las instrucciones en [Ver los mensajes MQTT con el cliente MQTT de AWS IoT \(p. 65\)](#) y configure la consola para que se suscriba a los mensajes con el nombre del tema de **Tema de** utilizado en su lugar utilice el filtro de temas comodín de #.

5. Pruebe el comando.

Mientras monitorea el tema en el cliente de prueba de la consola de AWS IoT, vaya a su cliente y ejecute la línea de comandos curl que creó en el paso 3. Debería ver los mensajes de su cliente en la consola.

Temas MQTT

Los temas MQTT identifican AWS IoT mensajes. Los clientes identifican los mensajes que publican dando los nombres de temas a los mensajes. Los clientes identifican los mensajes a los que desean suscribirse (recibir) registrando un filtro de temas con AWS IoT Core. El agente de mensajes de utiliza nombres y filtros de temas para dirigir los mensajes de los clientes de publicación a los clientes de suscripción.

El agente de mensajes utiliza temas para identificar los mensajes enviados mediante MQTT y enviados mediante HTTP al [URL de mensajes HTTPS \(p. 95\)](#).

Aunque AWS IoT admite algunos [temas reservados del sistema \(p. 100\)](#), la mayoría de los temas de MQTT los crea y administra el diseñador del sistema. AWS IoT utiliza temas para identificar los mensajes que se reciben de los clientes de publicación y para seleccionar los mensajes que se van a enviar a los clientes suscriptores, tal y como se describe en las secciones siguientes. Antes de crear un espacio de nombres de temas para el sistema, consulte las características de los temas de MQTT para crear la jerarquía de nombres de temas que funcione mejor con su sistema de IoT.

Nombres de temas

Los nombres de los temas y los filtros de temas son cadenas codificadas por UTF-8. Pueden representar una jerarquía de información utilizando el carácter de barra diagonal (/) para separar los niveles de la jerarquía. Por ejemplo, este nombre de tema podría referirse a un sensor de temperatura en la sala 1:

- sensor/temperature/room1

En este ejemplo, también puede haber otros tipos de sensores en otras salas con nombres de temas como:

- sensor/temperature/room2
- sensor/humidity/room1
- sensor/humidity/room2

Note

Cuando considere los nombres de los temas para los mensajes de su sistema, tenga en cuenta:

- Los nombres de temas y los filtros de temas distinguen entre mayúsculas y minúsculas.

- Los nombres de los temas no deben contener información de identificación personal.
- Los nombres de tema que comienzan por \$ son [temas reservados \(p. 100\)](#) que debe utilizar únicamente AWS IoT Core.
- AWS IoT Core no puede enviar ni recibir mensajes entre Cuentas de AWS o regiones.

Para obtener más información sobre cómo diseñar los nombres de los temas y el espacio de nombres, consulte nuestro documento técnico, [Diseñar temas MQTT para AWS IoT Core](#).

Para ver ejemplos de cómo las aplicaciones pueden publicar mensajes y suscribirse a ellos, comience con [Introducción a AWS IoT Core \(p. 17\)](#) y [AWS IoT SDK de dispositivos, SDK móviles y AWS IoT Client de dispositivos \(p. 1274\)](#).

Important

El espacio de nombres del tema está limitado a una Cuenta de AWS y una Región. Por ejemplo, el `sensor/temp/room1` tema utilizado por una Cuenta de AWS en una región es distinto de `sensor/temp/room1` tema utilizado por el mismo AWS cuenta en otra región o utilizada por cualquier otra cuenta de AWS en cualquier región.

ARN de tema

Todos los ARN de temas (nombres de recursos de Amazon) tienen el siguiente formulario:

```
arn:aws:iot:aws-region:AWS-account-ID:topic/Topic
```

Por ejemplo, `arn:aws:iot:us-west-2:123EXAMPLE456:topic/application/topic/device/sensor` es un ARN para el tema `application/topic/device/sensor`.

Filtros de temas

Los clientes suscriptores registran filtros de temas con el agente de mensajes para especificar los temas de mensajes que el agente de mensajes debe enviarles. Un filtro de tema puede ser un único nombre de tema para suscribirse a un único nombre de tema o puede incluir caracteres comodín para suscribirse a varios nombres de tema al mismo tiempo.

Los clientes de publicación no pueden utilizar caracteres comodín en los nombres de tema que publican.

En la tabla siguiente se enumeran los caracteres comodín que se pueden utilizar en un filtro de temas.

Comodines de tema

Carácter comodín	Coincide	Notas
#	Todas las cadenas en y por debajo de su nivel en la jerarquía de temas.	Debe ser el último carácter del filtro de temas. Debe ser el único carácter en su nivel de jerarquía de temas. Se puede utilizar en un filtro de temas que también contiene el carácter comodín +.
+	Cualquier cadena en el nivel que contiene el carácter.	Debe ser el único carácter en su nivel de jerarquía de temas.

Carácter comodín	Coincide	Notas
		Se puede utilizar en varios niveles de un filtro de tema.

Uso de comodines con los ejemplos de nombres de tema de sensor anteriores:

- Una suscripción a `sensor/#` recibe los mensajes publicados en `sensor/`, `sensor/temperature` y `sensor/temperature/room1`, pero no los mensajes publicados en `Sensor`.
- Una suscripción a `sensor/+/room1` recibe mensajes publicados en `sensor/temperature/room1` y `sensor/humidity/room1`, pero no mensajes enviados a `sensor/temperature/room2` o `sensor/humidity/room2`.

ARN de filtro de temas

Todos los filtros de temas ARN (nombres de recursos de Amazon) tienen el siguiente formulario:

```
arn:aws:iot:aws-region:AWS-account-ID:topicfilter/TopicFilter
```

Por ejemplo, `arn:aws:iot:us-west-2:123EXAMPLE456:topicfilter/application/topic/+` es un ARN para filtrar los temas `application/topic/+sensor`.

Carga útil de mensajes MQTT

La carga útil del mensaje que se envía en los mensajes MQTT no está especificada por AWS IoT, a menos que sea para uno de los [the section called “Temas reservados” \(p. 100\)](#). Para adaptarse a las necesidades de su aplicación, le recomendamos que defina la carga útil de mensajes para sus temas dentro de las restricciones del [AWS IoT Core Service Quotas para protocolos](#).

El uso de un formato JSON para la carga útil de mensajes habilita la AWS IoT motor de reglas para analizar los mensajes y aplicarle consultas SQL. Si la aplicación no requiere el motor de reglas para aplicar consultas SQL a las cargas útiles de mensajes, puede utilizar cualquier formato de datos que requiera la aplicación. Para obtener información sobre las limitaciones y los caracteres reservados de un documento JSON utilizado en consultas SQL, consulte [Extensões JSON \(p. 620\)](#).

Para obtener más información sobre cómo diseñar los temas de MQTT y sus cargas útiles de mensajes correspondientes, consulte [Diseñar temas MQTT para AWS IoT Core](#).

Si un límite de tamaño de mensaje supera las cuotas de servicio, se traducirá en `CLIENT_ERROR` con motivo `PAYLOAD_LIMIT_EXCEEDED` «La carga útil de mensajes supera el límite de tamaño para el tipo de mensaje». Para obtener más información acerca del límite de tamaño de mensajes, consulte [AWS IoT Core límites y cuotas de corredor de mensajes](#).

Temas reservados

Los temas que comienzan por un signo de dólar (\$) están reservados para su uso por parte de AWS IoT. Puede suscribirse y publicar en estos temas reservados según lo permitan; sin embargo, no puede crear nuevos temas que comiencen por un signo de dólar. Las operaciones de publicación o suscripción no admitidas a temas reservados pueden dar como resultado que se termine la conexión.

Temas del modelo de activos

Tema	Operaciones de cliente permitidas	Descripción
<code>\$aws/sitewise/asset-models/assetModelId</code>	Suscribirse	AWS IoT SiteWise publica notificaciones de propiedades

Tema	Operaciones de cliente permitidas	Descripción
<code>assets/<i>assetId</i>/properties/<i>propertyId</i></code>		de activos en este tema. Para obtener más información, consulte Interacción con otros AWS Servicios en la AWS IoT SiteWise Guía del usuario de.

Temas de Device Defender

Estos mensajes admiten búferes de respuesta en formato de representación concisa de objetos binarios (CBOR) y JavaScript Object Notation (Notación de objetos JSON), dependiendo de la [*payload-format*](#) del tema.

<i>payload-format</i>	Tipo de datos de formato de respuesta
cbor	Concise Binary Object Representation (Representación concisa de objetos binarios, CBOR)
json	JavaScript Object Notation (Notación de objetos de JavaScript, JSON)

Para obtener más información, consulte [Envío de métricas desde dispositivos \(p. 1018\)](#).

Tema	Operaciones permitidas	Descripción
<code>\$aws/things/<i>thingName</i>/defender/metrics/<i>payload-format</i></code>	Publicación	Los agentes de Device Defender publican métricas en este tema. Para obtener más información, consulte Envío de métricas desde dispositivos (p. 1018) .
<code>\$aws/things/<i>thingName</i>/defender/metrics/<i>payload-format</i>/accepted</code>	Suscribirse	AWS IoT publica en este tema después de que un agente de Device Defender publique un mensaje correcto en <code>aws/things/<i>thingName</i>/defender/metrics/<i>payload-format</i></code> . Para obtener más información, consulte Envío de métricas desde dispositivos (p. 1018) .
<code>\$aws/things/<i>thingName</i>/defender/metrics/<i>payload-format</i>/rejected</code>	Suscribirse	AWS IoT publica en este tema después de que un agente de Device Defender publique un mensaje incorrecto en <code>\$aws/things/<i>thingName</i>/defender/metrics/<i>payload-format</i></code> . Para obtener más información, consulte Envío de métricas desde dispositivos (p. 1018) .

Temas de eventos

Tema	Operaciones de cliente permitidas	Descripción
\$aws/events/certificates/registered/ <i>caCertificateId</i>	Suscribirse	AWS IoT publica este mensaje cuando AWS IoT registra automáticamente un certificado y cuando un cliente presenta un certificado con el estado PENDING_ACTIVATION. Para obtener más información, consulte the section called “Configurar la primera conexión de un cliente para el registro automático” (p. 311).
\$aws/events/job/ <i>JobId</i> /cancelado	Suscribirse	AWS IoT publica este mensaje cuando se cancela un trabajo. Para obtener más información, consulte Eventos de trabajos (p. 1108) .
\$aws/events/job/ <i>JobId</i> /cancellation_in_progress	Suscribirse	AWS IoT publica este mensaje cuando se está cancelando un trabajo. Para obtener más información, consulte Eventos de trabajos (p. 1108) .
\$aws/events/job/ <i>JobId</i> /completado	Suscribirse	AWS IoT publica este mensaje cuando se ha completado un trabajo. Para obtener más información, consulte Eventos de trabajos (p. 1108) .
\$aws/events/job/ <i>JobId</i> /eliminado	Suscribirse	AWS IoT publica este mensaje cuando se elimina un trabajo. Para obtener más información, consulte Eventos de trabajos (p. 1108) .
\$aws/events/job/ <i>JobId</i> /deletion_in_progress	Suscribirse	AWS IoT publica este mensaje cuando se está eliminando un trabajo. Para obtener más información, consulte Eventos de trabajos (p. 1108) .
\$aws/events/jobExecution/ <i>JobId</i> /cancelado	Suscribirse	AWS IoT publica este mensaje cuando se cancela la ejecución de un trabajo. Para obtener más información, consulte Eventos de trabajos (p. 1108) .
\$aws/events/jobExecution/ <i>JobId</i> /eliminado	Suscribirse	AWS IoT publica este mensaje cuando se elimina la ejecución de un trabajo. Para obtener

Tema	Operaciones de cliente permitidas	Descripción
		más información, consulte Eventos de trabajos (p. 1108) .
\$aws/events/jobExecution/ <i>JobId</i> /error	Suscribirse	AWS IoT publica este mensaje cuando se ha producido un error en la ejecución de un trabajo. Para obtener más información, consulte Eventos de trabajos (p. 1108) .
\$aws/events/jobExecution/ <i>JobId</i> /rejected	Suscribirse	AWS IoT publica este mensaje cuando se ha rechazado la ejecución de un trabajo. Para obtener más información, consulte Eventos de trabajos (p. 1108) .
\$aws/events/jobExecution/ <i>JobId</i> /removed	Suscribirse	AWS IoT publica este mensaje cuando se ha eliminado la ejecución de un trabajo. Para obtener más información, consulte Eventos de trabajos (p. 1108) .
\$aws/events/jobExecution/ <i>JobId</i> /exitoso	Suscribirse	AWS IoT publica este mensaje cuando la ejecución de un trabajo se ha realizado correctamente. Para obtener más información, consulte Eventos de trabajos (p. 1108) .
\$aws/events/jobExecution/ <i>JobId</i> /timed_out	Suscribirse	AWS IoT publica este mensaje cuando se ha agotado el tiempo de espera de ejecución de un trabajo. Para obtener más información, consulte Eventos de trabajos (p. 1108) .
\$aws/events/presence/connected/ <i>idCliente</i>	Suscribirse	AWS IoT publica en este tema cuando un cliente MQTT con el ID de cliente especificado se conecta a AWS IoT. Para obtener más información, consulte Eventos de conexión/desconexión (p. 1111) .
\$aws/events/presence/disconnected/ <i>idCliente</i>	Suscribirse	AWS IoT publica en este tema cuando un cliente MQTT con el ID de cliente especificado se desconecta de AWS IoT. Para obtener más información, consulte Eventos de conexión/desconexión (p. 1111) .

Tema	Operaciones de cliente permitidas	Descripción
\$aws/events/subscriptions/subscribed/ <i>idCliente</i>	Suscribirse	AWS IoT publica en este tema cuando un cliente MQTT con el ID de cliente especificado se suscribe a un tema MQTT. Para obtener más información, consulte Eventos de suscripción/cancelación de suscripción (p. 1114) .
\$aws/events/subscriptions/unsubscribed/ <i>idCliente</i>	Suscribirse	AWS IoT publica en este tema cuando un cliente MQTT con el ID de cliente especificado anula la suscripción de un tema MQTT. Para obtener más información, consulte Eventos de suscripción/cancelación de suscripción (p. 1114) .
\$aws/events/thing/ <i>thingName</i> /created	Suscribirse	AWS IoT publica en este tema cuando se crea el objeto <i>thingName</i> . Para obtener más información, consulte the section called “Eventos de registro” (p. 1100) .
\$aws/events/thing/ <i>thingName</i> /updated	Suscribirse	AWS IoT publica en este tema cuando se actualiza el objeto <i>thingName</i> . Para obtener más información, consulte the section called “Eventos de registro” (p. 1100) .
\$aws/events/thing/ <i>thingName</i> /deleted	Suscribirse	AWS IoT publica en este tema cuando se elimina el objeto <i>thingName</i> . Para obtener más información, consulte the section called “Eventos de registro” (p. 1100) .
\$aws/events/thingGroup/ <i>thingGroupName</i> /created	Suscribirse	AWS IoT publica en este tema cuando se crea el grupo de objetos, <i>thingGroupName</i> . Para obtener más información, consulte the section called “Eventos de registro” (p. 1100) .
\$aws/events/thingGroup/ <i>thingGroupName</i> /updated	Suscribirse	AWS IoT publica en este tema cuando se actualiza el grupo de objetos, <i>thingGroupName</i> . Para obtener más información, consulte the section called “Eventos de registro” (p. 1100) .

Tema	Operaciones de cliente permitidas	Descripción
\$aws/events/thingGroup/ <i>thingGroupName</i> /deleted	Suscribirse	AWS IoT publica en este tema cuando se elimina el grupo de objetos, <i>thingGroupName</i> . Para obtener más información, consulte the section called “Eventos de registro” (p. 1100) .
\$aws/events/thingType/ <i>thingTypeName</i> /created	Suscribirse	AWS IoT publica en este tema cuando se crea el tipo de objeto <i>thingTypeName</i> . Para obtener más información, consulte the section called “Eventos de registro” (p. 1100) .
\$aws/events/thingType/ <i>thingTypeName</i> /updated	Suscribirse	AWS IoT publica en este tema cuando se actualiza el tipo de objeto <i>thingTypeName</i> . Para obtener más información, consulte the section called “Eventos de registro” (p. 1100) .
\$aws/events/thingType/ <i>thingTypeName</i> /deleted	Suscribirse	AWS IoT publica en este tema cuando se elimina el tipo de objeto <i>thingTypeName</i> . Para obtener más información, consulte the section called “Eventos de registro” (p. 1100) .
\$aws/events/thingTypeAssociation/thing/ <i>thingName</i> / <i>thingTypeName</i>	Suscribirse	AWS IoT publica en este tema cuando el objeto, <i>thingName</i> , se asocia o desasocia del tipo de objeto, <i>thingTypeName</i> . Para obtener más información, consulte the section called “Eventos de registro” (p. 1100) .
\$aws/events/thingGroupMembership/thingGroup/ <i>thingGroupName</i> /thing/ <i>thingName</i> /added	Suscribirse	AWS IoT publica en este tema cuando el objeto, <i>thingName</i> , se agrega al grupo de objetos, <i>thingGroupName</i> . Para obtener más información, consulte the section called “Eventos de registro” (p. 1100) .

Tema	Operaciones de cliente permitidas	Descripción
\$aws/events/thingGroupMembership/thingGroup/ <i>thingGroupName</i> /thing/ <i>thingName</i> /removed	Suscribirse	AWS IoT publica en este tema cuando el objeto, <i>thingName</i> , se elimina del grupo de objetos, <i>thingGroupName</i> . Para obtener más información, consulte the section called “Eventos de registro” (p. 1100) .
\$aws/events/thingGroupHierarchy/thingGroup/ <i>parentThingGroupName</i> /childThingGroup/ <i>childThingGroupName</i> /added	Suscribirse	AWS IoT publica en este tema cuando el grupo de objetos, <i>childThingGroupName</i> , se agrega al grupo de objetos, <i>parentThingGroupName</i> . Para obtener más información, consulte the section called “Eventos de registro” (p. 1100) .
\$aws/events/thingGroupHierarchy/thingGroup/ <i>parentThingGroupName</i> /childThingGroup/ <i>childThingGroupName</i> /removed	Suscribirse	AWS IoT publica en este tema cuando el grupo de objetos, <i>childThingGroupName</i> , se elimina del grupo de objetos, <i>parentThingGroupName</i> . Para obtener más información, consulte the section called “Eventos de registro” (p. 1100) .

Temas de aprovisionamiento de flotas

Estos mensajes admiten búferes de respuesta en formato de representación concisa de objetos binarios (CBOR) y JavaScript Object Notation (Notación de objetos JSON), dependiendo de la *payload-format* del tema.

<i>payload-format</i>	Tipo de datos de formato de respuesta
cbor	Concise Binary Object Representation (Representación concisa de objetos binarios, CBOR)
json	JavaScript Object Notation (Notación de objetos de JavaScript, JSON)

Para obtener más información, consulte [API de MQTT de aprovisionamiento de dispositivos \(p. 819\)](#).

Tema	Operaciones de cliente permitidas	Descripción
\$aws/certificates/create/ <i>payload-format</i>	Publicación	Publique en este tema para crear un certificado a partir de una solicitud de firma de certificado (CSR).

Tema	Operaciones de cliente permitidas	Descripción
\$aws/certificates/create/ <i>payload-format</i> /accepted	Suscribirse	AWS IoT publica en este tema después de una llamada exitosa a \$aws/certificates/create/ <i>payload-format</i> .
\$aws/certificates/create/ <i>payload-format</i> /rejected	Suscribirse	AWS IoT publica en este tema después de una llamada fallida a \$aws/certificates/create/ <i>payload-format</i> .
\$aws/certificates/create-from-csr/ <i>payload-format</i>	Publicación	Publica en este tema para crear un certificado a partir de una CSR.
\$aws/certificates/create-from-csr/ <i>payload-format</i> /accepted	Suscribirse	AWS IoT publica en este tema una llamada exitosa a \$aws/certificates/create-from-csr/ <i>payload-format</i> .
\$aws/certificates/create-from-csr/ <i>payload-format</i> /rejected	Suscribirse	AWS IoT publica en este tema una llamada fallida a \$aws/certificates/create-from-csr/ <i>payload-format</i> .
\$aws/events/presence/connected/ <i>idCliente</i>	Suscribirse	AWS IoT publica en este tema cuando un cliente MQTT con el ID de cliente especificado se conecta a AWS IoT. Para obtener más información, consulte Eventos de conexión/desconexión (p. 1111) .
\$aws/provisioning-templates/ <i>templateName</i> /provision/ <i>payload-format</i>	Publicación	Publique en este tema para registrar un objeto.
\$aws/provisioning-templates/ <i>templateName</i> /provision/ <i>payload-format</i> /accepted	Suscribirse	AWS IoT publica en este tema después de una llamada exitosa a \$aws/provisioning-templates/ <i>TemplateName</i> /provision/ <i>payload-format</i> .
\$aws/provisioning-templates/ <i>templateName</i> /provision/ <i>payload-format</i> /rejected	Suscribirse	AWS IoT publica en este tema después de una llamada fallida a \$aws/provisioning-templates/ <i>TemplateName</i> /provision/ <i>payload-format</i> .

Temas de trabajos

Note

Las operaciones del cliente señaladas como Recibieren esta tabla indican los temas que AWS IoT publica directamente en el cliente que lo solicitó, independientemente de que el cliente se haya suscrito al tema o no. Los clientes deberían esperar recibir estos mensajes de respuesta incluso si no se han suscrito a ellos.

Estos mensajes de respuesta no pasan por el agente de mensajes y otros clientes o reglas no pueden suscribirlos. Para suscribirse a mensajes relacionados con la actividad del trabajo, utilice `lanotifynotify-nexttemas`.

Al suscribirse al trabajo y `jobExecution` temas de eventos para su solución de monitoreo de flotas, primero debe habilitar [eventos de trabajo y de ejecución de trabajos \(p. 1097\)](#) para recibir cualquier evento en el lado de la nube.

Para obtener más información, consulte [API de MQTT de dispositivo de trabajo \(p. 746\)](#).

Tema	Operaciones de cliente permitidas	Descripción
<code>\$aws/things/<i>thingName</i>/jobs/get</code>	Publicación	Los dispositivos publican un mensaje en este tema para realizar una solicitud <code>GetPendingJobExecutions</code> . Para obtener más información, consulte API de MQTT de dispositivo de trabajo (p. 746) .
<code>\$aws/things/<i>thingName</i>/jobs/get/accepted</code>	Suscribirse, recibir	Los dispositivos se suscriben a este tema para recibir respuestas correctas de una solicitud <code>GetPendingJobExecutions</code> . Para obtener más información, consulte API de MQTT de dispositivo de trabajo (p. 746) .
<code>\$aws/things/<i>thingName</i>/jobs/get/rejected</code>	Suscribirse, recibir	Los dispositivos se suscriben a este tema para recibir una respuesta cuando <code>unGetPendingJobExecutions</code> La solicitud se rechaza. Para obtener más información, consulte API de MQTT de dispositivo de trabajo (p. 746) .
<code>\$aws/things/<i>thingName</i>/jobs/start-next</code>	Publicación	Los dispositivos publican un mensaje en este tema para realizar una solicitud <code>StartNextPendingJobExecution</code> . Para obtener más información, consulte API de MQTT de dispositivo de trabajo (p. 746) .
<code>\$aws/things/<i>thingName</i>/jobs/start-next/accepted</code>	Suscribirse, recibir	Los dispositivos se suscriben a este tema para recibir respuestas correctas a una solicitud <code>StartNextPendingJobExecution</code> . Para obtener más información, consulte API de MQTT de dispositivo de trabajo (p. 746) .

Tema	Operaciones de cliente permitidas	Descripción
\$aws/things/ <i>thingName</i> /jobs/start-next/rejected	Suscribirse, recibir	Los dispositivos se suscriben a este tema para recibir una respuesta cuando unStartNextPendingJobExecutionLa solicitud se rechaza. Para obtener más información, consulte API de MQTT de dispositivo de trabajo (p. 746) .
\$aws/things/ <i>thingName</i> /jobs/ <i>jobId</i> /get	Publicación	Los dispositivos publican un mensaje en este tema para realizar una solicitud DescribeJobExecution. Para obtener más información, consulte API de MQTT de dispositivo de trabajo (p. 746) .
\$aws/things/ <i>thingName</i> /jobs/ <i>jobId</i> /get/accepted	Suscribirse, recibir	Los dispositivos se suscriben a este tema para recibir respuestas correctas a una solicitud DescribeJobExecution. Para obtener más información, consulte API de MQTT de dispositivo de trabajo (p. 746) .
\$aws/things/ <i>thingName</i> /jobs/ <i>jobId</i> /get/rejected	Suscribirse, recibir	Los dispositivos se suscriben a este tema para recibir una respuesta cuando unDescribeJobExecutionLa solicitud se rechaza. Para obtener más información, consulte API de MQTT de dispositivo de trabajo (p. 746) .
\$aws/things/ <i>thingName</i> /jobs/ <i>jobId</i> /update	Publicación	Los dispositivos publican un mensaje en este tema para realizar una solicitud UpdateJobExecution. Para obtener más información, consulte API de MQTT de dispositivo de trabajo (p. 746) .

Tema	Operaciones de cliente permitidas	Descripción
\$aws/things/ <i>thingName</i> /jobs/ <i>jobId</i> /update/accepted	Suscribirse, recibir	<p>Los dispositivos se suscriben a este tema para recibir respuestas correctas a una solicitud <code>UpdateJobExecution</code>. Para obtener más información, consulte API de MQTT de dispositivo de trabajo (p. 746).</p> <p>Nota</p> <p>Solo el dispositivo que publica en \$aws/things/<i>thingName</i>/jobs/<i>jobId</i>/update recibirá mensajes en este tema.</p>
\$aws/things/ <i>thingName</i> /jobs/ <i>jobId</i> /update/rejected	Suscribirse, recibir	<p>Los dispositivos se suscriben a este tema para recibir una respuesta cuando <code>unUpdateJobExecution</code> la solicitud se rechaza. Para obtener más información, consulte API de MQTT de dispositivo de trabajo (p. 746).</p> <p>Nota</p> <p>Solo el dispositivo que publica en \$aws/things/<i>thingName</i>/jobs/<i>jobId</i>/update recibirá mensajes en este tema.</p>
\$aws/things/ <i>thingName</i> /jobs/notify	Suscribirse	<p>Los dispositivos se suscriben a este tema para recibir notificaciones cuando la ejecución de un trabajo se añade o elimina de la lista de ejecuciones pendientes de un objeto. Para obtener más información, consulte API de MQTT de dispositivo de trabajo (p. 746).</p>

Tema	Operaciones de cliente permitidas	Descripción
\$aws/things/ <i>thingName</i> /jobs/notify-next	Suscribirse	Los dispositivos se suscriben a este tema para recibir notificaciones cuando cambia la siguiente ejecución de un trabajo pendiente para el objeto. Para obtener más información, consulte API de MQTT de dispositivo de trabajo (p. 746) .
\$aws/events/job/ <i>jobId</i> /completed	Suscribirse	El servicio de trabajos publica un evento sobre este tema cuando se completa un trabajo. Para obtener más información, consulte Eventos de trabajos (p. 1108) .
\$aws/events/job/ <i>jobId</i> /canceled	Suscribirse	El servicio de trabajos publica un evento sobre este tema cuando se cancela un trabajo. Para obtener más información, consulte Eventos de trabajos (p. 1108) .
\$aws/events/job/ <i>jobId</i> /deleted	Suscribirse	El servicio de trabajos publica un evento sobre este tema cuando se elimina un trabajo. Para obtener más información, consulte Eventos de trabajos (p. 1108) .
\$aws/events/job/ <i>jobId</i> /cancellation_in_progress	Suscribirse	El servicio de trabajos publica un evento sobre este tema cuando se inicia la cancelación de un trabajo. Para obtener más información, consulte Eventos de trabajos (p. 1108) .
\$aws/events/job/ <i>jobId</i> /deletion_in_progress	Suscribirse	El servicio de trabajos publica un evento sobre este tema cuando se inicia la eliminación de un trabajo. Para obtener más información, consulte Eventos de trabajos (p. 1108) .
\$aws/events/jobExecution/ <i>jobId</i> /succeeded	Suscribirse	El servicio de trabajos publica un evento sobre este tema cuando la ejecución del trabajo se ejecuta satisfactoriamente. Para obtener más información, consulte Eventos de trabajos (p. 1108) .

Tema	Operaciones de cliente permitidas	Descripción
\$aws/events/jobExecution/ <i>jobId</i> /failed	Suscribirse	El servicio de trabajos publica un evento sobre este tema cuando la ejecución de un trabajo produce un error. Para obtener más información, consulte Eventos de trabajos (p. 1108) .
\$aws/events/jobExecution/ <i>jobId</i> /rejected	Suscribirse	El servicio de trabajos publica un evento sobre este tema cuando se rechaza una ejecución de un trabajo. Para obtener más información, consulte Eventos de trabajos (p. 1108) .
\$aws/events/jobExecution/ <i>jobId</i> /canceled	Suscribirse	El servicio de trabajos publica un evento sobre este tema cuando se cancela la ejecución de un trabajo. Para obtener más información, consulte Eventos de trabajos (p. 1108) .
\$aws/events/jobExecution/ <i>jobId</i> /timed_out	Suscribirse	El servicio de trabajos publica un evento sobre este tema cuando se agota el tiempo de espera de ejecución de un trabajo. Para obtener más información, consulte Eventos de trabajos (p. 1108) .
\$aws/events/jobExecution/ <i>jobId</i> /removed	Suscribirse	El servicio de trabajos publica un evento sobre este tema cuando se elimina la ejecución de un trabajo. Para obtener más información, consulte Eventos de trabajos (p. 1108) .
\$aws/events/jobExecution/ <i>jobId</i> /deleted	Suscribirse	El servicio de trabajos publica un evento sobre este tema cuando se elimina la ejecución de un trabajo. Para obtener más información, consulte Eventos de trabajos (p. 1108) .

Temas de reglas

Tema	Operaciones de cliente permitidas	Descripción
\$aws/rules/ <i>nombreRegla</i>	Publicación	Los dispositivos o las aplicaciones publican en este tema para activar reglas directamente. Para

Tema	Operaciones de cliente permitidas	Descripción
		obtener más información, consulte Reducción de costos de mensajería con Basic Ingest (p. 560) .

Temas de tunelización segura

Tema	Operaciones de cliente permitidas	Descripción
\$aws/things/ <i>thing-name</i> /tunnels/notify	Suscribirse	AWS IoT publica este mensaje para que un agente de IoT inicie un proxy local en el dispositivo remoto. Para obtener más información, consulte the section called "Fragmento de agente de IoT" (p. 784) .

Temas de sombra

Las sombras con nombre y sin nombre utilizan los temas de esta sección. Los temas utilizados por cada uno solo difieren en el prefijo del tema. Esta tabla muestra el prefijo de tema utilizado por cada tipo de sombra.

Valor <i>ShadowTopicPrefix</i>	Tipo de sombra
\$aws/things/ <i>thingName</i> /shadow	Sombra sin nombre (clásica)
\$aws/things/ <i>thingName</i> /shadow/name/ <i>shadowName</i>	Sombra con nombre

Para crear un tema completo, seleccione la *ShadowTopicPrefix* para el tipo de sombra al que desea hacer referencia, sustituya *thingName*, si procede, *shadowName*, con sus valores correspondientes y, a continuación, anexe el código auxiliar del tema como se muestra en la tabla siguiente. Recuerde que los temas distinguen entre mayúsculas y minúsculas.

Tema	Operaciones de cliente permitidas	Descripción
<i>ShadowTopicPrefix</i> /delete	Publicar/suscribirse	Un dispositivo o una aplicación publica en este tema para eliminar una sombra. Para obtener más información, consulte /delete (p. 663) .
<i>ShadowTopicPrefix</i> /delete/accepted	Suscribirse	El servicio Device Shadow envía mensajes a este tema cuando se elimina una sombra. Para obtener más información, consulte /delete/accepted (p. 664) .

Tema	Operaciones de cliente permitidas	Descripción
<i>ShadowTopicPrefix/delete/rejected</i>	Suscribirse	El servicio Device Shadow envía mensajes a este tema cuando se rechaza una solicitud para eliminar una sombra. Para obtener más información, consulte /delete/rejected (p. 665) .
<i>ShadowTopicPrefix/get</i>	Publicar/suscribirse	Una aplicación o un objeto publica un mensaje vacío en este tema para obtener una sombra. Para obtener más información, consulte Temas MQTT de sombra de dispositivo (p. 657) .
<i>ShadowTopicPrefix/get/accepted</i>	Suscribirse	El servicio Device Shadow envía mensajes a este tema cuando se realiza correctamente una solicitud de una sombra. Para obtener más información, consulte /get/accepted (p. 658) .
<i>ShadowTopicPrefix/get/rejected</i>	Suscribirse	El servicio Device Shadow envía mensajes a este tema cuando se rechaza una solicitud de una sombra. Para obtener más información, consulte /get/rejected (p. 659) .
<i>ShadowTopicPrefix/update</i>	Publicar/suscribirse	Un objeto o una aplicación publica en este tema para actualizar una sombra. Para obtener más información, consulte /update (p. 660) .
<i>ShadowTopicPrefix/update/accepted</i>	Suscribirse	El servicio Device Shadow envía mensajes a este tema cuando se realiza correctamente una actualización en una sombra. Para obtener más información, consulte /update/accepted (p. 661) .
<i>ShadowTopicPrefix/update/rejected</i>	Suscribirse	El servicio Device Shadow envía mensajes a este tema cuando se rechaza una actualización en una sombra. Para obtener más información, consulte /update/rejected (p. 663) .

Tema	Operaciones de cliente permitidas	Descripción
<i>ShadowTopicPrefix/</i> update/delta	Suscribirse	El servicio Device Shadow envía mensajes a este tema cuando se detecta una diferencia entre las secciones para los estados reported y desired de una sombra. Para obtener más información, consulte /update/delta (p. 661) .
<i>ShadowTopicPrefix/</i> update/documents	Suscribirse	AWS IoT publica un documento de estado en este tema siempre que se realiza una actualización correcta de la sombra. Para obtener más información, consulte /update/documents (p. 662) .

Temas de entrega de archivos basados en MQTT

Estos mensajes admiten búferes de respuesta en formato de representación concisa de objetos binarios (CBOR) y JavaScript Object Notation (Notación de objetos JSON), dependiendo de la *payload-format* del tema.

<i>payload-format</i>	Tipo de datos de formato de respuesta
cbor	Concise Binary Object Representation (Representación concisa de objetos binarios, CBOR)
json	JavaScript Object Notation (Notación de objetos de JavaScript, JSON)

Tema	Operaciones de cliente permitidas	Descripción
\$aws/things/ <i>ThingName</i> / transmisiones/ <i>StreamId</i> / datos/ <i>payload-format</i>	Suscribirse	AWS La entrega de archivos basada en MQTT se publica en este tema si se acepta la solicitud «GetStream» de un dispositivo. La carga útil contiene los datos de transmisión. Para obtener más información, consulte Uso de AWS IoT Entrega de archivos basada en MQTT en dispositivos (p. 863) .
\$aws/things/ <i>ThingName</i> / transmisiones/ <i>StreamId</i> / get/ <i>payload-format</i>	Publicación	Un dispositivo publica en este tema para realizar una solicitud «GetStream». Para obtener más información, consulte Uso de AWS

Tema	Operaciones de cliente permitidas	Descripción
		IoTEntrega de archivos basada en MQTT en dispositivos (p. 863).
\$aws/things/ <i>ThingName</i> /transmisiones/ <i>StreamId</i> /descripción/ <i>payload-format</i>	Suscribirse	AWSLa entrega de archivos basada en MQTT se publica en este tema si se acepta la solicitud «DescribeStream» de un dispositivo. La carga útil contiene la descripción de la transmisión. Para obtener más información, consulte Uso deAWS IoTEntrega de archivos basada en MQTT en dispositivos (p. 863) .
\$aws/things/ <i>ThingName</i> /transmisiones/ <i>StreamId</i> /describe/ <i>payload-format</i>	Publicación	Un dispositivo publica en este tema para realizar una solicitud de «DescribeStream». Para obtener más información, consulte Uso deAWS IoTEntrega de archivos basada en MQTT en dispositivos (p. 863) .
\$aws/things/ <i>ThingName</i> /transmisiones/ <i>StreamId</i> /rechazado/ <i>payload-format</i>	Suscribirse	AWSLa entrega de archivos basada en MQTT se publica en este tema si se rechaza una solicitud «DescribeStream» o «GetStream» de un dispositivo. Para obtener más información, consulte Uso deAWS IoTEntrega de archivos basada en MQTT en dispositivos (p. 863) .

ARN de tema reservado

Todos los ARN del tema reservado (nombres de recursos de Amazon) tienen el siguiente formulario:

```
arn:aws:iot:aws-region:AWS-account-ID:topic/Topic
```

Por ejemplo, `arn:aws:iot:us-west-2:123EXAMPLE456:topic/$aws/things/thingName/jobs/get/accepted` es un ARN para el tema reservado `$aws/things/thingName/jobs/get/accepted`.

Puntos de enlace configurables

Note

Esta característica no está disponible en GovCloudRegiones de AWS.

conAWS IoT Core, puede configurar y administrar los comportamientos de los endpoints de datos mediante configuraciones de dominio. Puede generar variosAWS IoT Corepuntos finales de datos y

también personalice los endpoints de datos con sus propios nombres de dominio completos (y certificados de servidor asociados) y autorizadores. Para obtener más información acerca de los autorizadores personalizados, consulte [the section called “Autenticación personalizada” \(p. 319\)](#).

AWS IoT Core utiliza la extensión TLS de indicación de nombre de servidor (SNI) para aplicar configuraciones de dominio. Los dispositivos deben utilizar esta extensión cuando se conectan. También deben pasar un nombre de servidor idéntico al nombre de dominio especificado en la configuración de dominio.* Para probar este servicio, utilice la versión v2 del [AWS IoT SDKs de dispositivos](#) en GitHub.

Note

Si crea varios puntos finales de datos en su cuenta de AWS, compartirán AWS IoT Core recursos tales como temas de MQTT, sombras de dispositivos y reglas.

Puede utilizar configuraciones de dominio para simplificar tareas como las siguientes.

- Migrar dispositivos a AWS IoT
- Admitir flotas de dispositivos heterogéneos manteniendo configuraciones de dominio diferentes para cada tipo de dispositivo
- Mantener la identidad de la marca (por ejemplo, a través del nombre de dominio) migrando al mismo tiempo la infraestructura de la aplicación a AWS IoT

Puede configurar un nombre de dominio completo (FQDN) y también el certificado de servidor asociado. También puede asociar un autorizador personalizado. Para obtener más información, consulte [Autenticación personalizada \(p. 319\)](#).

Note

AWS IoT utiliza la extensión TLS de indicación de nombre de servidor (SNI) para aplicar configuraciones de dominio. Los dispositivos deben usar esta extensión al conectarse y pasar un nombre de servidor idéntico al nombre de dominio especificado en la configuración del dominio. Para probar este servicio, use la versión v2 de cada [SDK de dispositivo de AWS IoT en GitHub](#).

Note

Cuando proporciona los certificados de servidor para AWS IoT configuración de dominio personalizada, los certificados tienen un máximo de cuatro nombres de dominio. Para obtener más información, consulte [Puntos de enlace y cuotas de AWS IoT Core](#).

Temas

- [Creación y configuración de dominios administrados por AWS \(p. 117\)](#)
- [Creación y configuración de dominios personalizados \(p. 118\)](#)
- [Administración de configuraciones de dominio \(p. 121\)](#)

Creación y configuración de dominios administrados por AWS

Un punto de enlace configurable en un dominio administrado por AWS se crea con la API [CreateDomainConfiguration](#). Una configuración de dominio para un dominio administrado por AWS consta de lo siguiente:

- `domainConfigurationName`: un nombre definido por el usuario que identifica la configuración del dominio.

Note

Los nombres de configuración de dominio que comienzan por `IoT`: están reservados para los puntos de enlace predeterminados y no se pueden usar. Además, este valor debe ser exclusivo de región de AWS.

- `defaultAuthorizerName`(opcional): nombre del autorizador personalizado que se va a utilizar en el endpoint.
- `allowAuthorizerOverride`: valor booleano que especifica si los dispositivos pueden anular el autorizador predeterminado especificando otro autorizador en el encabezado HTTP de la solicitud. Este valor es obligatorio si se especifica un valor para `defaultAuthorizerName`.
- `serviceType`-AWS IoT actualmente solo admite el tipo de servicio. Cuando especificas `DATA`, AWS IoT devuelve un punto de enlace del tipo `iot:Data-ATS`. No se puede crear un punto de enlace configurable `iot:Data` (VeriSign).

El siguiente comando de la AWS CLI crea la configuración de dominio para un punto de enlace Data.

```
aws iot create-domain-configuration --domain-configuration-name "myDomainConfigurationName"  
--service-type "DATA"
```

Creación y configuración de dominios personalizados

Las configuraciones de dominio permiten especificar un nombre de dominio completo (FQDN) personalizado para conectarse a AWS IoT. En los dominios personalizados, puede administrar sus propios certificados de servidor y los detalles, como la entidad de certificación raíz (CA) que se utiliza para firmar el certificado, el algoritmo de firma, la profundidad de la cadena de certificados y el ciclo de vida del certificado.

El flujo de trabajo para establecer una configuración de dominio con un dominio personalizado consta de las tres etapas siguientes.

1. Registro de certificados de servidor en AWS Certificate Manager (p. 118)
2. Creación de una configuración de dominio (p. 119)
3. Creación de registros DNS (p. 120)

Registro de certificados de servidor en AWS Certificate Manager

Antes de crear una configuración de dominio con un dominio personalizado, debe registrar la cadena de certificados de servidor en [AWS Certificate Manager\(ACM\)](#). Puede utilizar tres tipos de certificados de servidor.

- [Certificados públicos generados por ACM \(p. 119\)](#)
- [Certificados externos firmados por una entidad emisora de certificación pública \(p. 119\)](#)
- [Certificados externos firmados por una entidad emisora de certificación privada \(p. 119\)](#)

Note

AWS IoT considera que un certificado está firmado por una entidad de certificación pública (CA) si está incluido en el paquete de CA de confianza de Mozilla.

Requisitos certificados

Consulte [Requisitos previos para la importación de certificados](#) para conocer los requisitos para importar certificados en ACM. Además de estos requisitos, AWS IoT Core añade los siguientes requisitos.

- El certificado de hoja debe incluir el uso de claves ampliadas o extensión x509 v3 con un valor deAuth del servidor(Autenticación del servidor web TLS). Si solicita el certificado a ACM, esta extensión se agrega automáticamente.
- La profundidad máxima de la cadena de certificados es de 5 certificados.

- El tamaño máximo de la cadena de certificados es de 16 KB.

Uso de un certificado para varios dominios

Si tiene previsto utilizar un certificado para cubrir varios subdominios, utilice un dominio comodín en el campo Nombre común (CN) o Nombres alternativos del firmante (SAN). Por ejemplo, utilice `*.iot.example.com` para cubrir dev.iot.example.com, qa.iot.example.com y prod.iot.example.com. Cada FQDN requiere su propia configuración de dominio, pero varias configuraciones de dominio pueden usar el mismo valor comodín. El CN o el SAN deben cubrir el FQDN que desea utilizar como dominio personalizado. Si hay SAN presentes, se ignora la CN y una SAN debe cubrir el FQDN que desea utilizar como dominio personalizado. Esta cobertura puede ser una coincidencia exacta o una coincidencia de caracteres comodín.

En las siguientes secciones se describe cómo obtener cada tipo de certificado. Cada recurso de certificado requiere un nombre de recurso de Amazon (ARN) registrado en ACM que utiliza al crear la configuración de dominio.

Certificados públicos generados por ACM

Puede generar un certificado público para el dominio personalizado mediante la API `RequestCertificate`. Cuando genera un certificado de esta manera, ACM valida que usted es el propietario del dominio personalizado. Para obtener más información, consulte [Solicitud de un certificado público](#) en la Guía del usuario de AWS Certificate Manager.

Certificados externos firmados por una entidad emisora de certificación pública

Si ya tiene un certificado de servidor firmado con una CA pública (una CA incluida en el paquete de CA de confianza de Mozilla), puede importar la cadena de certificados directamente a ACM mediante la `ImportCertificate` API. Para obtener más información sobre esta tarea, los requisitos previos y los requisitos de formato de certificado, consulte [Importación de certificados](#).

Certificados externos firmados por una entidad emisora de certificación privada

Si ya tiene un certificado de servidor firmado por una entidad emisora de certificación privada o autofirmado, puede utilizar el certificado para crear la configuración de dominio, pero también debe crear un certificado público adicional en ACM para validar que usted es el propietario del dominio. Para ello, registre la cadena de certificados de servidor en ACM mediante `ImportCertificate` API. Para obtener más información sobre esta tarea, los requisitos previos y los requisitos de formato de certificado, consulte [Importación de certificados](#).

Después de importar el certificado a ACM, genere un certificado público para el dominio personalizado mediante la `RequestCertificate` API. Cuando genera un certificado de esta manera, ACM valida que usted es el propietario del dominio personalizado. Para obtener más información, consulte [Solicitar un certificado público](#). Cuando cree la configuración de dominio, utilice este certificado público como certificado de validación.

Creación de una configuración de dominio

Un punto de enlace configurable en un dominio personalizado mediante la `CreateDomainConfiguration` API. Una configuración de dominio para un dominio personalizado consta de lo siguiente:

- `domainConfigurationName`: un nombre definido por el usuario que identifica la configuración del dominio.

Note

Los nombres de configuración de dominio que comienzan por `IoT`: están reservados para los puntos de enlace predeterminados y no se pueden usar. Además, este valor debe ser exclusivo deRegión de AWS.

- `domainName`— El FQDN al que utilizan sus dispositivos para conectarse a AWS IoT.

Note

AWS IoT utiliza la extensión TLS de indicación de nombre de servidor (SNI) para aplicar configuraciones de dominio. Los dispositivos deben usar esta extensión al conectarse y pasar un nombre de servidor idéntico al nombre de dominio especificado en la configuración del dominio.

- `serverCertificateArns`— El ARN de la cadena de certificados de servidor registrada en ACM. AWS IoT Core actualmente solo admite un certificado de servidor.
- `validationCertificateArn`— El ARN del certificado público que ha generado en ACM para validar la propiedad de su dominio personalizado. Este argumento no es necesario si utiliza un certificado de servidor firmado públicamente o uno generado por ACM.
- `defaultAuthorizerName`(opcional): nombre del autorizador personalizado que se va a utilizar en el endpoint.
- `allowAuthorizerOverride`: valor booleano que especifica si los dispositivos pueden anular el autorizador predeterminado especificando otro autorizador en el encabezado HTTP de la solicitud. Este valor es obligatorio si se especifica un valor para `defaultAuthorizerName`.
- `serviceType`—AWS IoT actualmente solo admite el tipo de servicio. Cuando especificas `DATA`, AWS IoT devuelve un punto de enlace del tipo `iot:Data-ATS`.

El siguiente comando de la AWS CLI crea una configuración de dominio 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

Después de crear la configuración de dominio, puede tardar hasta 60 minutos en AWS IoT servir los certificados de servidor personalizados.

Creación de registros DNS

Después de registrar la cadena de certificados de servidor y crear la configuración de dominio, cree un registro DNS para que el dominio personalizado apunte a un dominio de AWS IoT. Este registro debe apuntar a un punto de enlace de AWS IoT de tipo `iot:Data-ATS`. Puede obtener su punto de enlace mediante la [DescribeEndpoint API](#).

Los siguientes ejemplos de AWS CLI muestran cómo obtener su punto de enlace.

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

Cuando tenga el punto de enlace `iot:Data-ATS`, cree un registro `CNAME` en el dominio personalizado para este punto de enlace de AWS IoT. Si crea varios dominios personalizados en el mismo cuenta de AWS, alias a este mismo `iot:Data-ATS`.

Solución de problemas

Si tienes problemas para conectar dispositivos a un dominio personalizado, asegúrate de que AWS IoT Core ha aceptado y aplicado su certificado de servidor. Puedes verificar que AWS IoT Core ha aceptado su certificado mediante la [AWS IoT Core consola](#) o [AWS CLI](#).

Para utilizar la [AWS IoT Core consola](#), vaya a Configuración y seleccione el nombre de configuración del dominio. En el navegador Detalles del certificado de servidor, compruebe el estado y los detalles de

estado. Si el certificado no es válido, sustitúyalo en ACM por un certificado que cumpla la [requisitos de certificado \(p. 118\)](#) enumerados en la sección anterior. Si el certificado tiene el mismo ARN, AWS IoT Core lo recogerán y lo aplicarán automáticamente.

Para comprobar el estado del certificado mediante la AWS CLI, llama a [DescribeDomainConfiguration](#) API y especifique el nombre de configuración de dominio.

Note

Si el certificado no es válido, AWS IoT Core seguirá presentando el último certificado válido.

Puede comprobar qué certificado se entrega en el endpoint mediante el siguiente comando openssl.

```
openssl s_client -connect custom-domain-name:8883 -showcerts -servername custom-domain-name
```

Administración de configuraciones de dominio

Puede administrar los ciclos de vida de las configuraciones existentes mediante las siguientes API.

- [ListDomainConfigurations](#)
- [DescribeDomainConfiguration](#)
- [UpdateDomainConfiguration](#)
- [DeleteDomainConfiguration](#)

Consulta de las configuraciones de dominio

Usar [ListDomainConfigurations](#) API para devolver una lista paginada de todas las configuraciones de dominio de cuenta de AWS. Puede ver los detalles de una configuración de dominio en particular mediante la API [DescribeDomainConfiguration](#). Esta API toma un único parámetro `domainConfigurationName` y devuelve los detalles de la configuración especificada.

Actualización de configuraciones de dominio

Para actualizar el estado o el autorizador personalizado de su configuración de dominio, utilice la API [UpdateDomainConfiguration](#). Puede establecer el estado en `ENABLED` o `DISABLED`. Si deshabilita la configuración del dominio, los dispositivos conectados a ese dominio recibirán un error de autenticación.

Note

Actualmente no puede actualizar el certificado de servidor en la configuración de su dominio. Para cambiar el certificado de una configuración de dominio, debe eliminarlo y volver a crearlo.

Eliminación de configuraciones de dominio

Antes de eliminar una configuración de dominio, utilice la API [UpdateDomainConfiguration](#) para establecer el estado en `DISABLED`. Esto le ayuda a evitar que se elimine el punto de enlace por error. Después de deshabilitar la configuración del dominio, elimínela mediante la API [DeleteDomainConfiguration](#).

Note

Debes colocar AWS-dominios administrados en `DISABLED` estado durante 7 días antes de poder eliminarlos. Puede colocar dominios personalizados en `DISABLED` status y, a continuación, eliminarlos inmediatamente.

Después de eliminar una configuración de dominio, AWS IoT Core ya no sirve el certificado de servidor asociado a ese dominio personalizado.

Rotación de certificados en dominios personalizados

Es posible que tenga que reemplazar periódicamente el certificado de servidor por un certificado actualizado. La tasa a la que realiza esto dependerá del período de validez del certificado. Si ha generado el certificado de servidor mediante AWS Certificate Manager(ACM), puede configurar el certificado para que se renueve automáticamente. Cuando ACM renueva el certificado, AWS IoT Core recoge automáticamente el nuevo certificado. No es necesario realizar ninguna acción adicional. Si ha importado el certificado de servidor de otro origen, puede rotarlo importándolo de nuevo a ACM. Para obtener más información acerca de cómo volver a importar certificados, consulte [Volver a importar un certificado](#).

Note

AWS IoT Coresolo recoge las actualizaciones de certificados en las condiciones que se describen a continuación.

- El nuevo certificado tiene el mismo ARN que el antiguo.
- El nuevo certificado tiene el mismo algoritmo de firma, nombre común o nombre alternativo de sujeto que el anterior.

Conexión aAWS IoT Puntos de enlace FIPS

AWS IoT proporciona puntos finales que admiten el [Estándar de procesamiento de la información federal \(FIPS, Federal Information Processing Standard\) 140-2](#). Los puntos finales compatibles con FIPS son diferentes de los estándares AWS Puntos de enlace de . Para interactuar con AWS IoT de forma compatible con FIPS, debe utilizar los puntos finales que se describen a continuación con su cliente compatible con FIPS. La AWS IoT consola no cumple con FIPS.

En las secciones siguientes se describe cómo tener acceso a las normas FIPS AWS IoT los puntos de enlace mediante la API REST, un SDK de o la AWS CLI.

AWS IoT Core- Puntos de enlace del plano de control

Conformidad con FIPS AWS IoT Core- plano de control endpoints que admiten el AWS IoT operaciones y sus conexos Comandos de la CLI dese enumeran en [Puntos de enlace de FIPS por servicio](#). En [Puntos de enlace de FIPS por servicio](#), busque el AWS IoT Core- plano de control servicio y busque el punto final de su Región de AWS.

Para utilizar el endpoint compatible con FIPS cuando accede al AWS IoT operaciones, utilice el AWSSDK o la API REST con el punto final adecuado para su Región de AWS.

Para utilizar el endpoint compatible con FIPS cuando ejecuta aws iot Comandos de la CLI de, agregue el --endpoint parámetro con el punto final adecuado para su Región de AWS al comando de.

AWS IoT Core- Puntos de enlace del plano de datos

Conformidad con FIPS AWS IoT Core- plano de datos Puntos de enlace se enumeran en [Puntos de enlace de FIPS por servicio](#). En [Puntos de enlace de FIPS por servicio](#), busque el AWS IoT Core- plano de datos servicio y busque el punto final de su Región de AWS.

Puede utilizar el endpoint compatible con FIPS para su Región de AWS con un cliente compatible con FIPS mediante el AWS IoT SDK del dispositivo y proporcionar el punto final a la función de conexión del SDK en lugar de la predeterminada de su cuenta AWS IoT Core- plano de datos. La función de conexión es específica de la AWS IoT SDK de dispositivos. Para ver un ejemplo de una función de conexión, consulte la [Función de conexión en el AWS IoT Device SDK for Python](#).

Note

AWS IoT no admite la cuenta de AWS específica AWS IoT Core- plano de datos endpoints que cumplen con FIPS. Funciones de servicio que requieren una cuenta de AWS punto de enlace específico en la [Indicación de nombre de servidor \(SNI\) \(p. 382\)](#). Las no se pueden usar. Conformidad con FIPS AWS IoT Core- plano de datos los endpoints no pueden admitir [Certificados de registro de varias cuentas \(p. 299\)](#), [Dominios personalizados \(p. 118\)](#), y [Autorizadores personalizados \(p. 319\)](#).

AWS IoT Device Management- puntos finales de datos de trabajos

Conformidad con FIPS AWS IoT Device Management- datos de trabajo Puntos de enlace se enumeran en [Puntos de enlace de FIPS por servicio](#). En [Puntos de enlace de FIPS por servicio](#), busque el AWS IoT Device Management- datos de trabajo servicio y busque el punto final de su Región de AWS.

Para utilizar las normas FIPS AWS IoT Device Management- datos de trabajo endpoint cuando ejecutas [aws iot-jobs-data Comandos de la CLI de](#), agregue el --endpoint parámetro con el punto final adecuado para su Región de AWS al comando de. También puede usar la API REST con este punto de enlace.

Puede utilizar el endpoint compatible con FIPS para su Región de AWS con un cliente compatible con FIPS mediante el AWS IoT SDK del dispositivo y proporcionar el punto final a la función de conexión del SDK en lugar de la predeterminada de su cuenta AWS IoT Device Management- datos de trabajo. La función de conexión es específica de la AWS IoT SDK de dispositivos. Para ver un ejemplo de una función de conexión, consulte la [Función de conexión en el AWS IoT Device SDK for Python](#).

AWS IoT Device Management- Puntos finales de Fleet Hub

Conformidad con FIPS AWS IoT Device Management- Fleet Hub Puntos de enlace para utilizar con [Fleet Hub para AWS IoT Administración de dispositivos Comandos de la CLI de](#) se enumeran en [Puntos de enlace de FIPS por servicio](#). En [Puntos de enlace de FIPS por servicio](#), busque el AWS IoT Device Management- Fleet Hub servicio y busque el punto final de su Región de AWS.

Para utilizar las normas FIPS AWS IoT Device Management- Fleet Hub endpoint cuando ejecutas [aws iotfleethub Comandos de la CLI de](#), agregue el --endpoint parámetro con el punto final adecuado para su Región de AWS al comando de. También puede usar la API REST con este punto de enlace.

AWS IoT Device Management- puntos de enlace de túnel seguros

Conformidad con FIPS AWS IoT Device Management- tunelización segura Puntos de enlace de para el AWS IoT API de tunelización segura y el correspondiente [Comandos de la CLI de](#) se enumeran en [Puntos de enlace de FIPS por servicio](#). En [Puntos de enlace de FIPS por servicio](#), busque el AWS IoT Device Management- tunelización segura servicio y busque el punto final de su Región de AWS.

Para utilizar las normas FIPS AWS IoT Device Management- tunelización segura endpoint cuando ejecutas [aws iotsecuretunneling Comandos de la CLI de](#), agregue el --endpoint parámetro con el punto final adecuado para su Región de AWS al comando de. También puede usar la API REST con este punto de enlace.

Tutoriales de AWS IoT

LaAWS IoTlos tutoriales se dividen en dos vías de aprendizaje para apoyar dos objetivos diferentes. Elige la mejor ruta de aprendizaje para tu objetivo.

- Quieres construir un proof-of-concept para probar o demostrar unAWS IoTidea de solución

Para demostrar tareas y aplicaciones comunes de IoT mediante elAWS IoTDevice Client en tus dispositivos, sigue la[the section called “Creación de demostraciones con elAWS IoTClient del dispositivo” \(p. 124\)](#) ruta de aprendizaje. LaAWS IoTDevice Client proporciona software de dispositivo con el que puede aplicar sus propios recursos en la nube para demostrar un end-to-end solución con un desarrollo mínimo.

Para obtener más información acerca de laAWS IoTDevice Client, consulte la[AWS IoTClient del dispositivo](#).

- Desea aprender a crear software de producción para implementar su solución

Para crear su propio software de solución que cumpla con sus requisitos específicos mediante unAWS IoTSDK del dispositivo, siga el[the section called “Creación de soluciones con elAWS IoTSDKs de dispositivos” \(p. 176\)](#) ruta de aprendizaje.

Para obtener más información acerca de los disponiblesAWS IoTSDKs de dispositivos, consulte[???](#) (p. 1274). Para obtener más información acerca de laAWSSDKs, consulte[Herramientas sobre las basarseAWS](#).

AWS IoTopciones de ruta de aprendizaje tutorial

- [Creación de demostraciones con elAWS IoTClient del dispositivo \(p. 124\)](#)
- [Creación de soluciones con elAWS IoTSDKs de dispositivos \(p. 176\)](#)

Creación de demostraciones con elAWS IoTClient del dispositivo

Los tutoriales de esta ruta de aprendizaje le guiarán por los pasos para desarrollar software de demostración mediante elAWS IoTClient del dispositivo. LaAWS IoTDevice Client proporciona software que se ejecuta en su dispositivo IoT para probar y demostrar aspectos de una solución IoT basada enAWS IoT.

El objetivo de estos tutoriales es facilitar la exploración y la experimentación para que puedas estar seguro de queAWS IoTes compatible con la solución antes de desarrollar el software del dispositivo.

Lo que aprenderás en estos tutoriales:

- Cómo preparar un Raspberry Pi para usarlo como dispositivo IoT conAWS IoT
- Cómo demostrarAWS IoTfunciones mediante elAWS IoTDevice Client en el dispositivo

En esta ruta de aprendizaje, instalará elAWS IoTDevice Client en tu propio Raspberry Pi y crea elAWS IoTrecursos en la nube para demostrar ideas de soluciones de IoT. Si bien los tutoriales de esta ruta

de aprendizaje muestran características mediante el uso de una Raspberry Pi, explican los objetivos y procedimientos para ayudarle a adaptarlos a otros dispositivos.

Requisitos previos para crear demostraciones con elAWS IoTCliente del dispositivo

En esta sección se describe lo que debe tener antes de iniciar los tutoriales en esta ruta de aprendizaje.

Para completar los tutoriales en esta ruta de aprendizaje, necesitará:

- Una Cuenta de AWS

Puede utilizar su existenteCuenta de AWS, si tienes uno, pero es posible que tengas que añadir funciones o permisos adicionales para usar elAWS IoTcaracterísticas que utilizan estos tutoriales.

Si necesita crear una nuevaCuenta de AWS, consulte[the section called “Configurar suCuenta de AWS” \(p. 18\).](#)

- Raspberry Pi o dispositivo IoT compatible

Los tutoriales utilizan un[Raspberry Pi](#)porque viene en diferentes factores de forma, es omnipresente y es un dispositivo de demostración relativamente económico. Los tutoriales se han probado en la[Raspberry Pi 3 Modelo B+](#), el[Raspberry Pi 4 Modelo B](#)y en una instancia de Amazon EC2 que ejecuta Ubuntu Server 20.04 LTS (HVM). Para utilizar elAWS CLl ejecute los comandos, le recomendamos que utilice la última versión del Raspberry Pi OS ([Raspberry Pi OS \(64 bits\)](#)o OS Lite). Las versiones anteriores del sistema operativo podrían funcionar, pero no lo hemos probado.

Note

Los tutoriales explican los objetivos de cada paso para ayudarle a adaptarlos al hardware de IoT en el que no los hemos probado; sin embargo, no describen específicamente cómo adaptarlos a otros dispositivos.

- Familiaridad con el sistema operativo del dispositivo IoT

En los pasos de estos tutoriales se supone que está familiarizado con el uso de comandos y operaciones básicos de Linux desde la interfaz de línea de comandos compatible con Raspberry Pi. Si no estás familiarizado con estas operaciones, tal vez quieras dedicarte más tiempo para completar los tutoriales.

Para completar estos tutoriales, ya debe comprender cómo:

- Realice operaciones básicas del dispositivo de forma segura, como ensamblar y conectar componentes, conectar el dispositivo a las fuentes de alimentación necesarias e instalar y quitar tarjetas de memoria.
- Cargue y descargue el software y los archivos del sistema en el dispositivo. Si el dispositivo no utiliza un dispositivo de almacenamiento extraíble, como una tarjeta microSD, tendrá que saber cómo conectarse al dispositivo y cargar y descargar el software del sistema y los archivos en el dispositivo.
- Connect tu dispositivo a las redes en las que planeas usarlo.
- Connect al dispositivo desde otro equipo mediante un terminal SSH o un programa similar.
- Utilice una interfaz de línea de comandos para crear, copiar, mover, cambiar el nombre y establecer los permisos de los archivos y directorios del dispositivo.
- Instale nuevos programas en el dispositivo.
- Transfiera archivos desde y hacia su dispositivo mediante herramientas como FTP o SCP.
- Un entorno de desarrollo y pruebas para su solución IoT

Los tutoriales describen el software y el hardware necesarios; sin embargo, los tutoriales suponen que podrá realizar operaciones que podrían no describirse explícitamente. Algunos ejemplos de este hardware y operaciones incluyen:

- Un equipo host local en el que descargar y almacenar archivos

Para el Raspberry Pi, suele ser un ordenador personal o portátil que puede leer y escribir en tarjetas de memoria microSD. El equipo host local debe:

- Conectarse a Internet.
- Tenga la[AWS CLI](#)instalada y configurada.
- Disponer de un navegador web que admite laAWSconsola de .
- Una forma de conectar el equipo host local al dispositivo para comunicarse con él, introducir comandos y transferir archivos

En Raspberry Pi, esto se hace a menudo utilizando SSH y SCP desde el equipo host local.

- Monitor y teclado para conectarse al dispositivo IoT

Estos pueden ser útiles, pero no son necesarios para completar los tutoriales.

- Una forma de que el equipo host local y los dispositivos IoT se conecten a Internet

Podría ser una conexión de red por cable o inalámbrica a un router o puerta de enlace que está conectado a Internet. El host local también debe poder conectarse a Raspberry Pi. Esto podría requerir que estén en la misma red de área local. Los tutoriales no pueden mostrarle cómo configurarlo para la configuración de su dispositivo o dispositivo en particular, pero muestran cómo puede probar esta conectividad.

- Acceso al enrutador de la red de área local para ver los dispositivos conectados

Para completar los tutoriales de esta ruta de aprendizaje, tendrás que poder encontrar la dirección IP de tu dispositivo IoT.

En una red de área local, esto se puede hacer accediendo a la interfaz de administración del router de red al que se conectan los dispositivos. Si puede asignar una dirección IP fija para su dispositivo en el router, puede simplificar la reconexión cada vez que se reinicie el dispositivo.

Si tienes un teclado y un monitor conectados al dispositivo,ifconfigpuede mostrar la dirección IP del dispositivo.

Si ninguna de estas opciones es una opción, tendrás que encontrar una forma de identificar la dirección IP del dispositivo después de cada vez que se reinicie.

Después de tener todos sus materiales, continúe[the section called “Preparación de los dispositivos para elAWS IoTCliente del dispositivo” \(p. 127\)](#).

Tutoriales en esta trayectoria de aprendizaje

- [Tutorial: Preparación de los dispositivos para elAWS IoTCliente del dispositivo \(p. 127\)](#)
- [Tutorial: Instalar y configurar elAWS IoTCliente del dispositivo \(p. 137\)](#)
- [Tutorial: Demostrar la comunicación de mensajes MQTT con elAWS IoTCliente del dispositivo \(p. 146\)](#)
- [Tutorial: Demostrar acciones remotas \(trabajos\) con elAWS IoTCliente del dispositivo \(p. 159\)](#)
- [Tutorial: Limpieza después de la ejecución del sistemaAWS IoTTutorial del cliente del dispositivo \(p. 169\)](#)

Tutorial: Preparación de los dispositivos para elAWS IoTCliente del dispositivo

Este tutorial le guiará por la inicialización de su Raspberry Pi para prepararlo para los tutoriales posteriores en esta ruta de aprendizaje.

El objetivo de este tutorial es instalar la versión actual del sistema operativo del dispositivo y asegurarse de que puede comunicarse con el dispositivo en el contexto de su entorno de desarrollo.

Para comenzar este tutorial:

- Haga que los artículos estén listados en [the section called “Requisitos previos para crear demostraciones con elAWS IoTCliente del dispositivo” \(p. 125\)](#) disponible y listo para su uso.

Este tutorial tarda unos 90 minutos en completarse.

Cuando haya terminado este tutorial:

- El dispositivo IoT tendrá un sistema operativo actualizado.
- El dispositivo IoT tendrá el software adicional que necesita para los tutoriales posteriores.
- Sabrá que su dispositivo tiene conectividad a Internet.
- Habrá instalado un certificado obligatorio en su dispositivo.

Después de completar este tutorial, el siguiente tutorial prepara el dispositivo para las demostraciones que utilizan elAWS IoTCliente del dispositivo.

Procedimientos de este tutorial

- [Paso 1: Instalación y actualización del sistema operativo del dispositivo \(p. 127\)](#)
- [Paso 2: Instala y verifica el software necesario en tu dispositivo \(p. 130\)](#)
- [Paso 3: Probar el dispositivo y guardar el certificado de CA de Amazon \(p. 133\)](#)

Paso 1: Instalación y actualización del sistema operativo del dispositivo

Los procedimientos de esta sección describen cómo inicializar la tarjeta microSD que la Raspberry Pi utiliza para su unidad de sistema. La tarjeta microSD de Raspberry Pi contiene su software de sistema operativo (SO) así como espacio para el almacenamiento de archivos de su aplicación. Si no utilizas un Raspberry Pi, sigue las instrucciones del dispositivo para instalar y actualizar el software del sistema operativo del dispositivo.

Después de completar esta sección, debería poder iniciar el dispositivo IoT y conectarse a él desde el programa terminal de su equipo host local.

Equipo requerido:

- Su entorno de pruebas y desarrollo local
- Un Raspberry Pi que o su dispositivo IoT se puede conectar a Internet
- Tarjeta de memoria microSD con al menos 8 GB de capacidad o almacenamiento suficiente para el sistema operativo y el software requerido.

Note

Al seleccionar una tarjeta microSD para estos ejercicios, elija una que sea lo más grande que sea necesario pero, lo más pequeña posible.

Una pequeña tarjeta SD será más rápida de realizar copias de seguridad y actualización. En Raspberry Pi, no necesitarás más de una tarjeta microSD de 8 GB para estos tutoriales. Si necesita más espacio para su aplicación específica, los archivos de imagen más pequeños que guarde en estos tutoriales pueden cambiar el tamaño del sistema de archivos en una tarjeta más grande para utilizar todo el espacio admitido de la tarjeta que elija.

Equipamiento opcional:

- Un teclado USB conectado al Raspberry Pi
- Un monitor HDMI y un cable para conectar el monitor al Raspberry Pi

Procedimientos de esta sección:

- [Cargue el sistema operativo del dispositivo en la tarjeta microSD \(p. 128\)](#)
- [Inicie su dispositivo IoT con el nuevo sistema operativo \(p. 129\)](#)
- [Connect el equipo host local al dispositivo \(p. 129\)](#)

Cargue el sistema operativo del dispositivo en la tarjeta microSD

Este procedimiento utiliza el equipo host local para cargar el sistema operativo del dispositivo en una tarjeta microSD.

Note

Si el dispositivo no utiliza un medio de almacenamiento extraíble para su sistema operativo, instale el sistema operativo siguiendo el procedimiento correspondiente a ese dispositivo y continúe [the section called “Inicie su dispositivo IoT con el nuevo sistema operativo” \(p. 129\)](#).

Para instalar el sistema operativo en su Raspberry Pi

1. En el equipo host local, descargue y descomprima la imagen del sistema operativo Raspberry Pi que desea utilizar. Las últimas versiones están disponibles en <https://www.raspberrypi.com/software/operating-systems/>

Elegir una versión de Raspberry Pi OS

En este tutorial se utiliza Raspberry Pi OS Liteversión porque es la versión más pequeña que admite estos tutoriales en esta ruta de aprendizaje. Esta versión del sistema operativo Raspberry Pi solo tiene una interfaz de línea de comandos y no tiene interfaz gráfica de usuario. Una versión del último sistema operativo Raspberry Pi con una interfaz gráfica de usuario también funcionará con estos tutoriales; sin embargo, los procedimientos descritos en esta ruta de aprendizaje utilizan únicamente la interfaz de línea de comandos para realizar operaciones en Raspberry Pi.

2. Inserte la tarjeta microSD en el equipo host local.
3. Con una herramienta de imagen de tarjetas SD, escriba el archivo de imagen del sistema operativo descomprimido en la tarjeta microSD.
4. Despues de escribir la imagen del sistema operativo Raspberry Pi en la tarjeta microSD:
 - a. Abra la partición BOOT de la tarjeta microSD en una ventana de línea de comandos o en una ventana del explorador de archivos.

- b. En la partición BOOT de la tarjeta microSD, en el directorio raíz, cree un archivo vacío denominado `ssh` sin extensión de archivo ni contenido. Esto indica a Raspberry Pi que active las comunicaciones SSH la primera vez que se inicia.
5. Expulsa la tarjeta microSD y quítala de forma segura del equipo host local.

Su tarjeta microSD está lista para [the section called “Inicie su dispositivo IoT con el nuevo sistema operativo” \(p. 129\)](#).

Inicie su dispositivo IoT con el nuevo sistema operativo

Este procedimiento instala la tarjeta microSD e inicia la Raspberry Pi por primera vez utilizando el sistema operativo descargado.

Para iniciar el dispositivo IoT con el nuevo sistema operativo

1. Con la alimentación desconectada del dispositivo, inserte la tarjeta microSD del paso anterior, [the section called “Cargue el sistema operativo del dispositivo en la tarjeta microSD” \(p. 128\)](#), en el Raspberry Pi.
2. Conecta el dispositivo a una red cableada.
3. Estos tutoriales interactuarán con su Raspberry Pi desde su equipo host local mediante un terminal SSH.

Si también quieres interactuar directamente con el dispositivo, puedes:

- a. Conecta un monitor HDMI a él para ver los mensajes de la consola de Raspberry Pi antes de poder conectar la ventana del terminal del equipo host local a la Raspberry Pi.
 - b. Conecta un teclado USB si quieres interactuar directamente con la Raspberry Pi.
 4. Conecta la alimentación a la Raspberry Pi y espera alrededor de un minuto para que se inicialice.
- Si tienes un monitor conectado a tu Raspberry Pi, puedes ver el proceso de inicio en él.
5. Averigua la dirección IP de tu dispositivo:
 - Si ha conectado un monitor HDMI a Raspberry Pi, la dirección IP aparece en los mensajes mostrados en el monitor
 - Si tiene acceso al router al que se conecta Raspberry Pi, puede ver su dirección en la interfaz de administración del router.

Cuando tenga la dirección IP de su Raspberry Pi, estará listo para [the section called “Connect el equipo host local al dispositivo” \(p. 129\)](#).

Connect el equipo host local al dispositivo

Este procedimiento utiliza el programa de terminal del equipo host local para conectarse a Raspberry Pi y cambiar su contraseña predeterminada.

Para conectar el equipo host local al dispositivo

1. En el equipo host local, abra el programa terminal SSH:
 - Windows: PuTTY
 - Linux/macOS: Terminal

Note

PuTTY no se instala automáticamente en Windows. Si no está en el equipo, tal vez tenga que descargarlo e instalarlo.

2. Connect el programa de terminal a la dirección IP de Raspberry Pi e inicie sesión con sus credenciales predeterminadas.

```
username: pi  
password: raspberry
```

3. Después de iniciar sesión en Raspberry Pi, cambie la contraseña delpiusuario.

```
passwd
```

Siga las instrucciones para cambiar la contraseña.

```
Changing password for pi.  
Current password: raspberry  
New password: YourNewPassword  
Retype new password: YourNewPassword  
passwd: password updated successfully
```

Después de tener el símbolo de línea de comandos de Raspberry Pi en la ventana del terminal y cambiar la contraseña, estará listo para continuar[the section called “Paso 2: Instala y verifica el software necesario en tu dispositivo” \(p. 130\)](#).

Paso 2: Instala y verifica el software necesario en tu dispositivo

Los procedimientos de esta sección continúan desde[la sección anterior \(p. 127\)](#) para actualizar el sistema operativo de Raspberry Pi e instalar el software en Raspberry Pi que se utilizará en la siguiente sección para compilar e instalar elAWS IoTClient del dispositivo.

Después de completar esta sección, su Raspberry Pi tendrá un sistema operativo actualizado, el software requerido por los tutoriales de esta ruta de aprendizaje y se configurará para su ubicación.

Equipo requerido:

- Su entorno de pruebas y desarrollo local desde[la sección anterior \(p. 127\)](#)
- La Raspberry Pi que usaste en[la sección anterior \(p. 127\)](#)
- La tarjeta de memoria microSD de[la sección anterior \(p. 127\)](#)

Note

Raspberry Pi Model 3+ y Raspberry Pi Model 4 pueden ejecutar todos los comandos descritos en esta ruta de aprendizaje. Si el dispositivo IoT no puede compilar software o ejecutar elAWS Command Line Interface, es posible que tenga que instalar los compiladores necesarios en el equipo host local para crear el software y, a continuación, transferirlo a su dispositivo IoT. Para obtener más información sobre cómo instalar y crear software para su dispositivo, consulte la documentación del software de su dispositivo.

Procedimientos de esta sección:

- [Actualizar el software del sistema operativo \(p. 131\)](#)

- Instale las aplicaciones y bibliotecas necesarias (p. 132)
- (Opcional) Guardar la imagen de la tarjeta microSD (p. 132)

Actualizar el software del sistema operativo

Este procedimiento actualiza el software del sistema operativo.

Para actualizar el software del sistema operativo en el dispositivo Raspberry Pi

Lleve a cabo estos pasos en la ventana de terminal del equipo host local.

1. Escriba estos comandos para actualizar el software del sistema en su Raspberry Pi.

```
sudo apt-get -y update
sudo apt-get -y upgrade
sudo apt-get -y autoremove
```

2. Actualice la configuración regional y de la zona horaria de Raspberry Pi (opcional).

Introduzca este comando para actualizar la configuración regional y la zona horaria del dispositivo.

```
sudo raspi-config
```

- a. Para configurar la configuración regional del dispositivo:

- i. En el navegadorHerramienta de configuración del software Raspberry Pi (raspi-config)pantalla, elija la opción5.

5 Localisation Options Configure language and regional settings

UsarTabclave para moverse<Select>,y luego pulsespace bar.

- ii. En el menú de opciones de localización, elija la opciónL1.

L1 Locale Configure language and regional settings

UsarTabclave para moverse<Select>,y luego pulsespace bar.

- iii. En la lista de opciones de configuración regional, elija las configuraciones regionales que desea instalar en su Raspberry Pi utilizando las teclas de flecha para desplazarse y elspace barpara marcar los que quieras.

En los Estados Unidos,**en_US.UTF-8**es bueno para elegir.

- iv. Después de seleccionar las configuraciones regionales de su dispositivo, utilice elTabclave para elegir<OK>y, a continuación, pulsespace barPara visualizarConfiguración de configuraciones regionalespágina de confirmación.

- b. Para configurar la zona horaria del dispositivo:

- i. En el navegadorraspi-configpantalla, elija la opción5.

5 Localisation Options Configure language and regional settings

UsarTabclave para moverse<Select>,y luego pulsespace bar.

- ii. En el menú de opciones de localización, utilice la tecla de flecha para elegir la opciónL2:

L2 time zone Configure time zone

UsarTabclave para moverse<Select>,y luego pulsespace bar.

- iii. En el navegadorConfiguración de tzdata, elija su área geográfica de la lista.

Usar Tabclave para moverse<OK>y, a continuación, pulsespace bar.

- iv. En la lista de ciudades, usa las teclas de flecha para elegir una ciudad de tu zona horaria.

Para configurar la zona horaria, utilice la opciónTabclave para moverse<OK>y, a continuación, pulsespace bar.

- c. Cuando haya terminado de actualizar la configuración, utilice elTabclave para moverse<Finish>y, a continuación, pulsespace barpara cerrarraspi-configapp.

3. Escriba este comando para reiniciar el dispositivo Raspberry Pi.

```
sudo shutdown -r 0
```

4. Espere a que se reinicie su Raspberry Pi.

5. Una vez reiniciado su Raspberry Pi, vuelva a conectar la ventana del terminal de su equipo host local a su Raspberry Pi.

El software del sistema Raspberry Pi ya está configurado y está listo para continuar[the section called “Instale las aplicaciones y bibliotecas necesarias” \(p. 132\)](#).

Instale las aplicaciones y bibliotecas necesarias

Este procedimiento instala el software de la aplicación y las bibliotecas que utilizan los tutoriales posteriores.

Si utiliza una Raspberry Pi o si puede compilar el software necesario en su dispositivo IoT, lleve a cabo estos pasos en la ventana del terminal del equipo host local. Si debe compilar software para su dispositivo IoT en el equipo host local, consulte la documentación de software de su dispositivo IoT para obtener información sobre cómo realizar estos pasos en el dispositivo.

Para instalar el software de aplicación y las bibliotecas en su Raspberry Pi

1. Introduzca este comando para instalar el software de la aplicación y las bibliotecas.

```
sudo apt-get -y install build-essential libssl-dev cmake unzip git python3-pip
```

2. Introduzca estos comandos para confirmar que se ha instalado la versión correcta del software.

```
gcc --version
cmake --version
openssl version
git --version
```

3. Confirme que estas versiones del software de la aplicación están instaladas:

- gcc: 9.3.0 o posterior
- cmake: 3.10.x o posterior
- OpenSSL: 1.1.1 o posterior
- git: 2.20.1 o posterior

Si su Raspberry Pi tiene versiones aceptables del software de aplicación requerido, está listo para continuar[the section called “\(Opcional\) Guardar la imagen de la tarjeta microSD” \(p. 132\)](#).

(Opcional) Guardar la imagen de la tarjeta microSD

A lo largo de los tutoriales de esta ruta de aprendizaje, encontrará estos procedimientos para guardar una copia de la imagen de la tarjeta microSD de Raspberry Pi en un archivo del equipo host local. Si bien

se les anima, no son tareas obligatorias. Al guardar la imagen de la tarjeta microSD donde se sugiere, puede omitir los procedimientos que preceden al punto de guardado en esta ruta de aprendizaje, lo que puede ahorrar tiempo si encuentra la necesidad de volver a intentar algo. La consecuencia de no guardar periódicamente la imagen de la tarjeta microSD es que es posible que tenga que reiniciar los tutoriales en la ruta de aprendizaje desde el principio si la tarjeta microSD está dañada o si configura accidentalmente una aplicación o sus ajustes incorrectamente.

En este punto, la tarjeta microSD de su Raspberry Pi tiene un sistema operativo actualizado y el software básico de aplicación cargado. Puede ahorrar el tiempo que le tomó completar los pasos anteriores guardando ahora el contenido de la tarjeta microSD en un archivo. Tener la imagen actual de la imagen de la tarjeta microSD de tu dispositivo te permite comenzar desde este punto para continuar o volver a intentar un tutorial o procedimiento sin necesidad de instalar y actualizar el software desde cero.

Para guardar la imagen de la tarjeta microSD en un archivo

1. Introduzca este comando para apagar la Raspberry Pi.

```
sudo shutdown -h 0
```

2. Despues de que la Raspberry Pi se apague por completo, elimine su potencia.
3. Retire la tarjeta microSD de la Raspberry Pi.
4. En el equipo host local:
 - a. Inserta la tarjeta microSD.
 - b. Con la herramienta de imagen de la tarjeta SD, guarda la imagen de la tarjeta microSD en un archivo.
 - c. Una vez guardada la imagen de la tarjeta microSD, expulsa la tarjeta del equipo host local.
5. Con la alimentación desconectada de la Raspberry Pi, inserta la tarjeta microSD en la Raspberry Pi.
6. Aplique energía al Raspberry Pi.
7. Despues de esperar aproximadamente un minuto, en el equipo host local, vuelva a conectar la ventana del terminal del equipo host local que estaba conectado a Raspberry Pi. y, a continuación, inicie sesión en Raspberry Pi.

Paso 3: Probar el dispositivo y guardar el certificado de CA de Amazon

Los procedimientos de esta sección continúan desdela sección anterior (p. 130)para instalar elAWS Command Line Interfacey el certificado de entidad de certificación utilizado para autenticar sus conexiones conAWS IoT Core.

Después de completar esta sección, sabrá que su Raspberry Pi tiene el software del sistema necesario para instalar elAWS IoTDevice Client y que tiene una conexión funcional a Internet.

Equipo requerido:

- Su entorno de pruebas y desarrollo local desdela sección anterior (p. 130)
- La Raspberry Pi que usaste enla sección anterior (p. 130)
- La tarjeta de memoria microSD dela sección anterior (p. 130)

Procedimientos de esta sección:

- [Instalar la AWS Command Line Interface \(p. 134\)](#)
- [Configuración de sus credenciales de Cuenta de AWS \(p. 134\)](#)

- Descargar el certificado de entidad de certificación Amazon Root (p. 135)
- (Opcional) Guardar la imagen de la tarjeta microSD (p. 136)

Instalar la AWS Command Line Interface

Este procedimiento instala elAWS CLl en su Raspberry Pi.

Si utiliza una Raspberry Pi o si puede compilar software en su dispositivo IoT, lleve a cabo estos pasos en la ventana del terminal del equipo host local. Si debe compilar software para su dispositivo IoT en el equipo host local, consulte la documentación de software de su dispositivo IoT para obtener información sobre las bibliotecas que necesita.

Para instalar elAWS CLl en su Raspberry Pi

1. Ejecute estos comandos para descargar e instalar elAWS 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. Ejecute este comando para instalar elAWS CLI. Este comando puede tardar hasta 15 minutos en completarse.

```
pip3 install . # install the AWS CLI
```

3. Ejecute este comando para confirmar que la versión correcta delAWS CLI se instaló.

```
aws --version
```

La versión delAWS CLI debe ser 2.2 o posterior.

Si el archivo deAWS CLl muestra su versión actual, estará listo para continuar [the section called "Configuración de sus credenciales de Cuenta de AWS" \(p. 134\)](#).

Configuración de sus credenciales de Cuenta de AWS

En este procedimiento, obtendrá Cuenta de AWS credenciales y agréguelas para usarlas en su Raspberry Pi.

Para añadir Cuenta de AWS credenciales de tu dispositivo

1. Obtenga un Access Key ID y Clave de acceso secretas desde su Cuenta de AWS para autenticar elAWS CLl en su dispositivo.

Si es la primera vez que usa AWS IAM, <https://aws.amazon.com/premiumsupport/knowledge-center/create-access-key/> describe el proceso que se va a ejecutar en elAWS consola para crear AWS Credenciales de IAM para usar en el dispositivo.

2. En la ventana de terminal de su equipo host local que está conectado a su Raspberry Pi y con el Access Key ID y Clave de acceso secretas credenciales para tu dispositivo:

- a. Ejecute la AWS configure app con este comando:

```
aws configure
```

- b. Introduzca sus credenciales e información de configuración cuando se le solicite:

```
AWS Access Key ID: your Access Key ID
AWS Secret Access Key: your Secret Access Key
Default region name: your Región de AWS code
Default output format: json
```

3. Ejecute este comando para probar el acceso de tu dispositivo a tu cuenta de AWS IoT Core.

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

Debería devolver su cuenta de AWS específico AWS IoT punto final de datos, como este ejemplo:

```
{  
    "endpointAddress": "a3EXAMPLEffp-ats.iot.us-west-2.amazonaws.com"  
}
```

Si ves tu cuenta de AWS específico AWS IoT punto final de datos, tu Raspberry Pi tiene la conectividad y los permisos para continuar [the section called “Descargar el certificado de entidad de certificación Amazon Root” \(p. 135\)](#).

Important

Sus credenciales de cuenta de AWS se almacenan ahora en la tarjeta microSD de su Raspberry Pi. Si bien esto hace interacciones futuras con AWS fácil para ti y el software que crearás en estos tutoriales, también se guardarán y duplicarán en cualquier imagen de tarjeta microSD que hagas después de este paso de forma predeterminada.

Para proteger la seguridad de su cuenta de AWS credenciales, antes de guardar más imágenes de tarjeta microSD, considere borrar las credenciales ejecutando `aws configure` nuevo e introduciendo caracteres aleatorios para el Access Key ID y Clave de acceso secretas para evitar su cuenta de AWS credenciales de comprometidas.

Si descubre que ha guardado su cuenta de AWS credenciales sin darse cuenta, puede desactivarlas en la consola IAM.

Descargar el certificado de entidad de certificación Amazon Root

Este procedimiento descarga y guarda una copia de un certificado de la Autoridad de certificación raíz (CA) de Amazon. Al descargar este certificado se guarda para utilizarlo en los tutoriales posteriores y también prueba la conectividad de su dispositivo con AWS servicios de .

Para descargar y guardar el certificado de entidad de certificación de Amazon Root

- Ejecute este comando para crear un directorio para el certificado.

```
mkdir ~/certs
```

- Ejecute este comando para descargar el certificado de entidad de certificación de Amazon Root.

```
curl -o ~/certs/AmazonRootCA1.pem https://www.amazontrust.com/repository/
AmazonRootCA1.pem
```

- Ejecute estos comandos para establecer el acceso al directorio de certificados y a su archivo.

```
chmod 745 ~
chmod 700 ~/certs
chmod 644 ~/certs/AmazonRootCA1.pem
```

- Ejecute este comando para ver el archivo de certificado CA en el nuevo directorio.

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

Debería ver una entrada como esta. La fecha y la hora serán diferentes; sin embargo, el tamaño del archivo y el resto de la información deben ser los mismos que se muestran aquí.

```
-rw-r--r-- 1 pi pi 1188 Oct 28 13:02 AmazonRootCA1.pem
```

Si el tamaño del archivo no es 1188, consulte curl parámetros de comandos. Es posible que hayas descargado un archivo incorrecto.

(Opcional) Guardar la imagen de la tarjeta microSD

En este punto, la tarjeta microSD de su Raspberry Pi tiene un sistema operativo actualizado y el software básico de aplicación cargado.

Para guardar la imagen de la tarjeta microSD en un archivo

- En la ventana del terminal del equipo host local, borre elAWS Credenciales de .

- Ejecute la AWS configure app con este comando:

```
aws configure
```

- Sustituya sus credenciales cuando se le solicite. Puedes irte Nombre de región predeterminado y Formato de salida predeterminado tal como están presionando Entrar.

```
AWS Access Key ID [*****YT2H]: XYXXXXXX
AWS Secret Access Key [*****9plH]: XYXXXXXX
Default region name [us-west-2]:
Default output format [json]:
```

- Introduzca este comando para apagar la Raspberry Pi.

```
sudo shutdown -h 0
```

- Después de que la Raspberry Pi se apague por completo, retire su conector de alimentación.
- Retire la tarjeta microSD del dispositivo.
- En el equipo host local:
 - Inserta la tarjeta microSD.
 - Con la herramienta de imagen de la tarjeta SD, guarda la imagen de la tarjeta microSD en un archivo.
 - Una vez guardada la imagen de la tarjeta microSD, expulsa la tarjeta del equipo host local.
- Con la alimentación desconectada de la Raspberry Pi, inserta la tarjeta microSD en la Raspberry Pi.
- Aplique energía al dispositivo.
- Después de un minuto, en el equipo host local, reinicie la sesión de la ventana de terminal e inicie sesión en el dispositivo.

No vuelvas a ingresar tu cuenta de AWS credenciales todavía.

Después de reiniciar e iniciar sesión en su Raspberry Pi, estará listo para continuar the section called "Instalar y configurar el AWS IoT Cliente del dispositivo" (p. 137).

Tutorial: Instalar y configurar elAWS IoTClient del dispositivo

Este tutorial le guía a través de la instalación y configuración deAWS IoTDevice Client y creación deAWS IoTRecursos que utilizarás en esta y en otras demostraciones.

Para comenzar este tutorial:

- Haga que su equipo host local y Raspberry Pi deel tutorial anterior (p. 127)listo.

Este tutorial puede tardar hasta 90 minutos en completarse.

Cuando haya terminado de examinar este tema:

- El dispositivo IoT estará listo para usar en otrosAWS IoTDemostaciones del cliente de
- Habrás aprovisionado tu dispositivo IoT enAWS IoT Core.
- Habrás descargado e instalado elAWS IoTDevice Client en tu dispositivo.
- Habrás guardado una imagen de la tarjeta microSD de tu dispositivo que se puede utilizar en tutoriales posteriores.

Equipo requerido:

- Su entorno de pruebas y desarrollo local desde[la sección anterior \(p. 133\)](#)
- La Raspberry Pi que usaste en[la sección anterior \(p. 133\)](#)
- La tarjeta de memoria microSD de la Raspberry Pi que usaste en[la sección anterior \(p. 133\)](#)

Procedimientos de este tutorial

- [Paso 1: Descargue y guarde elAWS IoTClient del dispositivo \(p. 137\)](#)
- [\(Opcional\) Guardar la imagen de la tarjeta microSD \(p. 139\)](#)
- [Paso 2: Aprovisione su Raspberry Pi enAWS IoT \(p. 139\)](#)
- [Paso 3: Configuración delAWS IoTClient de dispositivo para probar la conectividad \(p. 143\)](#)

Paso 1: Descargue y guarde elAWS IoTClient del dispositivo

Los procedimientos de esta sección descargan elAWS IoTDevice Client, compílelo e instálelo en su Raspberry Pi. Después de probar la instalación, puedes guardar la imagen de la tarjeta microSD de Raspberry Pi para usarla más adelante cuando quieras volver a probar los tutoriales.

Procedimientos de esta sección:

- [Descargue y compruebe elAWS IoTClient del dispositivo \(p. 137\)](#)
- [Cree los directorios utilizados por los tutoriales \(p. 138\)](#)

Descargue y compruebe elAWS IoTClient del dispositivo

Este procedimiento instala elAWS IoTClient de dispositivos en su Raspberry Pi.

Realice estos comandos en la ventana de terminal del equipo host local que está conectado a Raspberry Pi.

Para instalar elAWS IoTClient del dispositivo en Raspberry Pi

1. Introduzca estos comandos para descargar y compilar elAWS IoTClient de dispositivos en su 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. Ejecute este comando para compilar elAWS IoTClient del dispositivo. Este comando puede tardar hasta 15 minutos en completarse.

```
cmake --build . --target aws-iot-device-client
```

Los mensajes de advertencia mostrados como AWS IoTLas compilaciones de Device Client se pueden ignorar.

Estos tutoriales se han probado con elAWS IoTClient de dispositivo basado en gcc, versión (Raspbian 10.2.1-6+rpi1) 10.2.1 20210110 en la versión 30 de octubre de 2021 de Raspberry Pi OS (bullseye) engcc, versión (Raspbian 8.3.0-6+rpi1) 8.3.0 en la versión 7 de mayo de 2021 del sistema operativo Raspberry Pi (buster).

3. Después delAWS IoTDevice Client finaliza la compilación, pruébalo ejecutando este comando.

```
./aws-iot-device-client --help
```

Si ve la ayuda de la línea de comandos deAWS IoTDevice Client, elAWS IoTDevice Client se ha creado correctamente y está listo para su uso.

Cree los directorios utilizados por los tutoriales

Este procedimiento crea los directorios en Raspberry Pi que se utilizarán para almacenar los archivos utilizados por los tutoriales en esta ruta de aprendizaje.

Para crear los directorios utilizados por los tutoriales en esta ruta de aprendizaje:

1. Ejecute estos comandos para crear los directorios necesarios.

```
mkdir ~/dc-configs  
mkdir ~/policies  
mkdir ~/messages  
mkdir ~/certs/testconn  
mkdir ~/certs/pubsub  
mkdir ~/certs/jobs
```

2. Ejecute estos comandos para establecer los permisos de los nuevos directorios.

```
chmod 745 ~  
chmod 700 ~/certs/testconn  
chmod 700 ~/certs/pubsub  
chmod 700 ~/certs/jobs
```

Después de crear estos directorios y establecer su permiso, continúe en[the section called “\(Opcional\) Guardar la imagen de la tarjeta microSD” \(p. 139\)](#).

(Opcional) Guardar la imagen de la tarjeta microSD

En este punto, la tarjeta microSD de su Raspberry Pi tiene un sistema operativo actualizado, el software básico de aplicación y el AWS IoT Cliente del dispositivo.

Si quieres volver a probar estos ejercicios y tutoriales, puedes omitir los procedimientos anteriores escribiendo la imagen de la tarjeta microSD que guardas con este procedimiento en una nueva tarjeta microSD y continuar con los tutoriales de [the section called “Paso 2: Aprovisione su Raspberry Pi enAWS IoT”](#) (p. 139).

Para guardar la imagen de la tarjeta microSD en un archivo:

En la ventana de terminal del equipo host local que está conectado a la Raspberry Pi:

1. Confirmar que su cuenta de AWS no se han almacenado las credenciales.

- a. Ejecute la AWS configure app con este comando:

```
aws configure
```

- b. Si se han almacenado sus credenciales (si se muestran en la solicitud), introduzca **laxXXXXXX** cadena cuando se le solicite tal y como se muestra aquí. Dejar Nombre de región predeterminado y Formato de salida predeterminado en blanco.

```
AWS Access Key ID [*****YXXX]: XXXXXXXX
AWS Secret Access Key [*****YXXX]: XXXXXXXX
Default region name:
Default output format:
```

2. Ingrese este comando para apagar el dispositivo Raspberry Pi.

```
sudo shutdown -h 0
```

3. Despues de que la Raspberry Pi se apague por completo, retire su conector de alimentación.
4. Retire la tarjeta microSD del dispositivo.
5. En el equipo host local:

- a. Inserta la tarjeta microSD.
- b. Con la herramienta de imagen de la tarjeta SD, guarda la imagen de la tarjeta microSD en un archivo.
- c. Una vez guardada la imagen de la tarjeta microSD, expulsa la tarjeta del equipo host local.

Puedes continuar con esta tarjeta microSD en [the section called “Paso 2: Aprovisione su Raspberry Pi enAWS IoT”](#) (p. 139).

Paso 2: Aprovisione su Raspberry Pi enAWS IoT

Los procedimientos de esta sección comienzan con la imagen microSD guardada que tiene el AWS CLI y AWS IoT Device Client instalado y crea el AWS IoT recursos y certificados de dispositivo que aprovisionan su Raspberry Pi en AWS IoT.

Instale la tarjeta microSD en su Raspberry Pi

Este procedimiento instala la tarjeta microSD con el software necesario cargado y configurado en Raspberry Pi y configura su cuenta de AWS para que puedas continuar con los tutoriales de esta ruta de aprendizaje.

Utilizar una tarjeta microSD desde [the section called “\(Opcional\) Guardar la imagen de la tarjeta microSD” \(p. 139\)](#) que cuenta con el software necesario para los ejercicios y tutoriales de esta trayectoria de aprendizaje.

Para instalar la tarjeta microSD en su Raspberry Pi

1. Con la alimentación desconectada de la Raspberry Pi, inserta la tarjeta microSD en la Raspberry Pi.
2. Aplique energía al Raspberry Pi.
3. Despues de un minuto, en el equipo host local, reinicie la sesión de la ventana de terminal e inicie sesión en Raspberry Pi.
4. En el equipo host local, en la ventana de terminal y con el Access Key ID y la clave de acceso secretas credenciales de su Raspberry Pi:
 - a. Ejecute la AWS configure app con este comando:

```
aws configure
```

- b. Escriba su cuenta de AWS credenciales e información de configuración cuando se le solicite:

```
AWS Access Key ID [*****YXXY]: your Access Key ID
AWS Secret Access Key [*****YXXY]: your Secret Access Key
Default region name [us-west-2]: your Región de AWS code
Default output format [json]: json
```

Después de haber restaurado su cuenta de AWS credenciales, está listo para continuar [the section called “Aprovisionamiento del dispositivo en AWS IoT Core” \(p. 140\)](#).

Aprovisionamiento del dispositivo en AWS IoT Core

Los procedimientos de esta sección crean el AWS IoT recursos que aprovisionan su Raspberry Pi en AWS IoT. A medida que crees estos recursos, se te pedirá que registres varios datos. Esta información la utiliza el AWS IoT Configuración del cliente de dispositivos en el siguiente procedimiento.

Para que tu Raspberry Pi funcione con AWS IoT, debe aprovisionarse. El aprovisionamiento es el proceso de creación y configuración de los AWS IoT recursos necesarios para admitir su Raspberry Pi como un dispositivo de IoT.

Con la Raspberry Pi encendida y reiniciada, conecte la ventana del terminal de su equipo host local al Raspberry Pi y complete estos procedimientos.

Procedimientos de esta sección:

- [Crear y descargar archivos de certificado de dispositivo \(p. 140\)](#)
- [Crear AWS IoT recursos \(p. 141\)](#)

Crear y descargar archivos de certificado de dispositivo

Este procedimiento crea los archivos de certificado de dispositivo para esta demostración.

Para crear y descargar los archivos de certificado de dispositivo para su Raspberry Pi

1. En la ventana de terminal del equipo host local, introduzca estos comandos para crear los archivos de certificado de dispositivo para su 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"
```

El comando devuelve una respuesta similar a la siguiente. Anote el **certificateArn** valor para un uso posterior.

```
{  
    "certificateArn": "arn:aws:iot:us-  
west-2:57EXAMPLE833:cert/76e7e4edb3e52f52334be2f387a06145b2aa4c7fc810f3aea2d92abc227d269",  
    "certificateId":  
    "76e7e4edb3e52f5233EXAMPLE7a06145b2aa4c7fc810f3aea2d92abc227d269",  
    "certificatePem": "-----BEGIN CERTIFICATE-----  
\nMIIDWTCCAkGgAwIBAgI_SHORTENED_FOR_EXAMPLE_Lgn4jfghtS\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" }  
}
```

- Introduzca los siguientes comandos para establecer los permisos en el directorio de certificados y sus archivos.

```
chmod 745 ~  
chmod 700 ~/certs/testconn  
chmod 644 ~/certs/testconn/*  
chmod 600 ~/certs/testconn/private.pem.key
```

- Ejecute este comando para revisar los permisos de los directorios y archivos de certificados.

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

El resultado del comando debe ser el mismo que el que ve aquí, excepto que las fechas y horas del archivo serán 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
```

En este punto, tiene instalados los archivos de certificado del dispositivo en su Raspberry Pi y puede continuar [the section called “CrearAWS IoT recursos” \(p. 141\)](#).

CrearAWS IoT recursos

Este procedimiento aprovisiona su dispositivo en AWS IoT creando los recursos a los que necesita acceder su dispositivo AWS IoT características y servicios.

Para aprovisionar el dispositivo en AWS IoT

- En la ventana de terminal de su equipo host local, introduzca el siguiente comando para obtener la dirección del punto final de datos del dispositivo para su cuenta de AWS.

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

El comando de los pasos anteriores devuelve una respuesta similar a la siguiente. Anote el `endpointAddress` valor para un uso posterior.

```
{  
    "endpointAddress": "a3qjEXAMPLEffp-ats.iot.us-west-2.amazonaws.com"  
}
```

2. Escriba este comando para crear unAWS IoT recurso de cosa para su Raspberry Pi.

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

Si las recetasAWS IoT objeto de recurso se creó, el comando devuelve una respuesta similar a esta.

```
{  
    "thingName": "DevCliTestThing",  
    "thingArn": "arn:aws:iot:us-west-2:57EXAMPLE833:thing/DevCliTestThing",  
    "thingId": "8ea78707-32c3-4f8a-9232-14bEXAMPLEfd"  
}
```

3. En la ventana de borna:

- Abra un editor de texto, comonano.
- Copie este documento de política JSON y péguelo en el editor de texto abierto.

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

Note

Este documento de política otorga generosamente permiso a todos los recursos para conectarse, recibir, publicar y suscribirse. Normalmente, las políticas otorgan permiso únicamente a recursos específicos para realizar acciones específicas. Sin embargo, para la prueba de conectividad inicial de dispositivos, esta política excesivamente general y permisiva se utiliza para minimizar la posibilidad de que se produzca un problema de acceso durante esta prueba. En los tutoriales posteriores, se utilizarán documentos de política con un alcance más limitado para demostrar mejores prácticas en el diseño de políticas.

- Guardé el archivo en el editor de texto como~/**policies/dev_cli_test_thing_policy.json**.
- Ejecute este comando para utilizar el documento de política de los pasos anteriores para crear unAWS IoT política.

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

Si se crea la política, el comando devolverá una respuesta similar a esta.

```
{  
    "policyName": "DevCliTestThingPolicy",  
    "policyArn": "arn:aws:iot:us-west-2:57EXAMPLE833:policy/DevCliTestThingPolicy",  
    "policyDocument": "{\n        \"Version\": \"2012-10-17\", \"Statement\": [\n            {\n                \"Effect\": \"Allow\", \"Action\": [\n                    \"iot:Publish\", \"iot:Subscribe\", \"iot:Receive\"\n                ], \"Resource\": [\n                    \"*\"\n                ]\n            }\n        ]\n    },\n    \"policyVersionId\": \"1\"\n}
```

- Ejecute este comando para asociar la política al certificado de dispositivo. Reemplazar *certificateArn* con *certificateArn* valor que guardaste anteriormente.

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

Si se ejecuta correctamente, este comando no devuelve nada.

- Ejecute este comando para adjuntar el certificado de dispositivo alAWS IoTRecurso de objeto. Reemplazar *certificateArn* con *certificateArn* valor que guardaste anteriormente.

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

Si se ejecuta correctamente, este comando no devuelve nada.

Después de aprovisionar correctamente el dispositivo enAWS IoT, está listo para continuar[the section called “Paso 3: Configuración delAWS IoTClient de dispositivo para probar la conectividad” \(p. 143\)](#).

Paso 3: Configuración delAWS IoTClient de dispositivo para probar la conectividad

Los procedimientos de esta sección configuran laAWS IoTDevice Client para publicar un mensaje de MQTT desde su Raspberry Pi.

Procedimientos de esta sección:

- [Crear el archivo de configuración \(p. 143\)](#)
- [Cliente de prueba MQTT abierto \(p. 145\)](#)
- [Ejecución deAWS IoTClient del dispositivo \(p. 145\)](#)

Crear el archivo de configuración

Este procedimiento crea el archivo de configuración para probar elAWS IoTClient del dispositivo.

Para crear el archivo de configuración para probar el AWS IoT Cliente del dispositivo

- En la ventana de terminal del equipo host local que está conectado a la Raspberry Pi:
 - a. Introduzca estos comandos para crear un directorio para los archivos de configuración y establecer el permiso en el directorio:

```
mkdir ~/dc-configs
chmod 745 ~/dc-configs
```

- b. Abra un editor de texto, comonano.
- c. Copie este documento JSON y péguelo en el editor de texto abierto.

```
{
    "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. Sustituya elvalor con endpoint de datos de dispositivo para suCuenta de AWSque encontraste enthe section called “Aprovisionamiento del dispositivo enAWS IoT Core” (p. 140).
- e. Guarde el archivo en el editor de texto como-/dc-configs/dc-testconn-config.json.
- f. Ejecute este comando para establecer los permisos en el nuevo archivo de configuración.

```
chmod 644 ~/dc-configs/dc-testconn-config.json
```

Después de guardar el archivo, estará listo para continuarthe section called “Cliente de prueba MQTT abierto” (p. 145).

Cliente de prueba MQTT abierto

Este procedimiento prepara elCliente de prueba MQTTeen laAWS IoTconsola para suscribirse al mensaje de MQTT de que elAWS IoTDevice Client se publica cuando se ejecuta.

Para preparar elCliente de prueba MQTTPara suscribirse a todos los mensajes MQTT

1. En el equipo host local, en elAWS IoTconsola, eligeCliente de prueba MQTT.
2. En el navegadorSuscripción a un temapestaña, enFiltro de temas, introduzca#(un signo de una libra) y elijaSuscribirsepara suscribirse a todos los temas de MQTT.
3. Debajo delSuscripcionesetiqueta, confirma que ves#(un signo de una libra).

Deja la ventana con elCliente de prueba MQTTabierto a medida que continúasthe section called “Ejecución deAWS IoTClient del dispositivo” (p. 145).

Ejecución deAWS IoTClient del dispositivo

Este procedimiento ejecuta elAWS IoTDevice Client para que publique un único mensaje MQTT que indicaCliente de prueba MQTTrecibe y muestra.

Para enviar un mensaje MQTT desde elAWS IoTClient del dispositivo

1. Asegúrese de que tanto la ventana de terminal conectada a su Raspberry Pi como la ventana con elCliente de prueba MQTTestán visibles mientras realiza este procedimiento.
2. En la ventana de terminal, introduzca estos comandos para ejecutar elAWS IoTClient de dispositivo mediante el archivo de configuración creado enthe section called “Crear el archivo de configuración” (p. 143).

```
cd ~/aws-iot-device-client/build
./aws-iot-device-client --config-file ~/dc-configs/dc-testconn-config.json
```

En una ventana de terminal, elAWS IoTDevice Client muestra mensajes de información y cualquier error que se produzca cuando se ejecuta.

Si no se muestran errores en la ventana del terminal, revise elCliente de prueba MQTT.

3. En el navegadorCliente de prueba MQTT, en la ventana Suscripciones, consulte la¡Hello World!mensaje enviado a latest/dc/pubtopic tema de mensaje.
4. Si el archivo deAWS IoTDevice Client no muestra errores y verá¡Hello World!enviado atest/dc/pubtopicmensaje en elCliente de prueba MQTT, ha demostrado una conexión correcta.
5. En una ventana de terminal, escriba^c(Ctrl-C) para detener elAWS IoTClient del dispositivo.

Después de demostrar que elAWS IoTDevice Client se ejecuta correctamente en su Raspberry Pi y puede comunicarse conAWS IoT, puede continuar hasta elthe section called “[Demostrar la comunicación de mensajes MQTT con elAWS IoTClient del dispositivo](#)” (p. 146).

Tutorial: Demostrar la comunicación de mensajes MQTT con elAWS IoTClient del dispositivo

En este tutorial se muestra cómo elAWS IoTDevice Client puede suscribirse y publicar mensajes MQTT, que se utilizan habitualmente en soluciones de IoT.

Para comenzar este tutorial:

- Haga que su equipo host local y Raspberry Pi estén configurados como se utilizan en[sección anterior \(p. 137\)](#).

Si ha guardado la imagen de la tarjeta microSD después de instalar elAWS IoTDevice Client, puedes usar una tarjeta microSD con esa imagen con tu Raspberry Pi.

- Si ya has ejecutado esta demo antes, revisa[??? \(p. 171\)](#)para eliminar todoAWS IoTRecursos que creó en ejecuciones anteriores para evitar errores de recursos duplicados.

Este tutorial tarda aproximadamente 45 minutos en completarse.

Cuando haya terminado de examinar este tema:

- Habrás demostrado diferentes formas en que tu dispositivo IoT puede suscribirse a mensajes MQTT desdeAWS IoT publicar mensajes MQTT enAWS IoT.

Equipo necesario:

- Su entorno de pruebas y desarrollo local desde[sección anterior \(p. 137\)](#)
- La Raspberry Pi que usaste en[sección anterior \(p. 137\)](#)
- La tarjeta de memoria microSD de la Raspberry Pi que usaste en[sección anterior \(p. 137\)](#)

Procedimientos de este tutorial

- [Paso 1: Preparación de Raspberry Pi para demostrar la comunicación de los mensajes MQTT \(p. 146\)](#)
- [Paso 2: Demostrar la publicación de mensajes con elAWS IoTClient del dispositivo \(p. 151\)](#)
- [Paso 3: Demostrar suscribirse a mensajes con elAWS IoTClient del dispositivo \(p. 154\)](#)

Paso 1: Preparación de Raspberry Pi para demostrar la comunicación de los mensajes MQTT

Este procedimiento crea los recursos enAWS IoT en Raspberry Pi para demostrar la comunicación de mensajes MQTT utilizando elAWS IoTClient del dispositivo.

Procedimientos de esta sección:

- [Cree los archivos de certificado para demostrar la comunicación MQTT \(p. 147\)](#)
- [Aprovisionamiento de su dispositivo para demostrar la comunicación MQTT \(p. 148\)](#)
- [Configuración delAWS IoTArchivo de configuración de Device Client y cliente de prueba MQTT para demostrar la comunicación MQTT \(p. 150\)](#)

Cree los archivos de certificado para demostrar la comunicación MQTT

Este procedimiento crea los archivos de certificado de dispositivo para esta demostración.

Para crear y descargar los archivos de certificado de dispositivo para su Raspberry Pi

1. En la ventana de terminal del equipo host local, introduzca el siguiente comando para crear los archivos de certificado de dispositivo para su 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"
```

El comando devuelve una respuesta similar a la siguiente. Save the **certificateArn** valor para un uso posterior.

```
{
  "certificateArn": "arn:aws:iot:us-
west-2:57EXAMPLE833:cert/76e7e4edb3e52f52334be2f387a06145b2aa4c7fc810f3aea2d92abc227d269",
  "certificateId": "76e7e4edb3e52f5233EXAMPLE7a06145b2aa4c7fc810f3aea2d92abc227d269",
  "certificatePem": "-----BEGIN CERTIFICATE-----
\nMIIDWTCCAkGgAwIBAgI_SHORTENED_FOR_EXAMPLE_Lgn4jfgtS\n-----END CERTIFICATE-----\n",
  "keyPair": {
    "PublicKey": "-----BEGIN PUBLIC KEY-----
\nMIIBIjANBggqhkiG9w0BA_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. Introduzca los siguientes comandos para establecer los permisos en el directorio de certificados y sus archivos.

```
chmod 700 ~/certs/pubsub
chmod 644 ~/certs/pubsub/*
chmod 600 ~/certs/pubsub/private.pem.key
```

3. Ejecute este comando para revisar los permisos de los directorios y archivos de certificados.

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

El resultado del comando debe ser el mismo que el que ve aquí, excepto que las fechas y horas del archivo serán 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. Introduzca estos comandos para crear los directorios de los archivos de registro.

```
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/*
```

Aprovisionamiento de su dispositivo para demostrar la comunicación MQTT

En esta sección se crea elAWS IoT recursos que aprovisionan su Raspberry Pi enAWS IoT.

Para aprovisionar el dispositivo enAWS IoT:

1. En la ventana de terminal del equipo host local, introduzca el siguiente comando para obtener la dirección del punto final de datos del dispositivo para suCuenta de AWS.

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

El valor del endpoint no ha cambiado desde el momento en que ejecutó este comando para el tutorial anterior. La ejecución de nuevo aquí se hace para facilitar la búsqueda y pegar el valor del extremo de datos en el archivo de configuración utilizado en este tutorial.

El comando de los pasos anteriores devuelve una respuesta similar a la siguiente. Registre el*endpointAddress* valor para un uso posterior.

```
{  
  "endpointAddress": "a3qjEXAMPLEffp-ats.iot.us-west-2.amazonaws.com"  
}
```

2. Escriba este comando para crear un nuevoAWS IoT recurso de cosa para su Raspberry Pi.

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

Porque unAWS IoT cosa recurso es una representación virtual de su dispositivo en la nube, podemos crear múltiples recursos enAWS IoT para utilizar con fines diferentes. Todas ellas pueden ser utilizadas por el mismo dispositivo IoT físico para representar diferentes aspectos del dispositivo.

Estos tutoriales solo usarán un recurso a la vez para representar la Raspberry Pi. De esta forma, en estos tutoriales, representan las diferentes demostraciones para que después de crear elAWS IoT recursos para una demostración, puede volver atrás y repetir la demostración utilizando los recursos que creó específicamente para cada uno.

Si las recetasAWS IoT se creó un recurso de la cosa, el comando devuelve una respuesta similar a esta.

```
{  
  "thingName": "PubSubTestThing",  
  "thingArn": "arn:aws:iot:us-west-2:57EXAMPLE833:thing/PubSubTestThing",  
  "thingId": "8ea78707-32c3-4f8a-9232-14bEXAMPLEfd"  
}
```

3. En la ventana de bornea:

- a. Abra un editor de texto, comonano.
- b. Copie este documento JSON y péguelo en el editor de texto abierto.

```
{  
  "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. En el editor, en cada sección de recursos del documento de política, sustituya **us-west-2:57EXAMPLE833** con tu Región de AWS, un carácter de dos puntos (:) y tus 12 dígitos Cuenta de AWS número.
 - d. Guarde el archivo en el editor de texto como **~/policies/pubsub_test_thing_policy.json**.
4. Ejecute este comando para utilizar el documento de política de los pasos anteriores para crear una AWS IoT política.

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

Si se crea la política, el comando devuelve una respuesta similar a esta.

```
{
    "policyName": "PubSubTestThingPolicy",
    "policyArn": "arn:aws:iot:us-west-2:57EXAMPLE833:policy/PubSubTestThingPolicy",
    "policyDocument": "{\n    \"Version\": \"2012-10-17\", \"Statement\": [\n        {\n            \"Effect\": \"Allow\", \"Action\": \"iot:Connect\", \"Resource\": \"arn:aws:iot:us-west-2:57EXAMPLE833:client/PubSubTestThing\"},\n            {\n                \"Effect\": \"Allow\", \"Action\": \"iot:Publish\", \"Resource\": \"arn:aws:iot:us-west-2:57EXAMPLE833:topic/test/dc/pubtopic\"},\n                {\n                    \"Effect\": \"Allow\", \"Action\": \"iot:Subscribe\", \"Resource\": \"arn:aws:iot:us-west-2:57EXAMPLE833:topicfilter/test/dc/subtopic\"},\n                    {\n                        \"Effect\": \"Allow\", \"Action\": \"iot:Receive\", \"Resource\": \"arn:aws:iot:us-west-2:57EXAMPLE833:topic/test/dc/subtopic\"}\n    ]\n}
```

```
"policyVersionId": "1"
```

5. Ejecute este comando para asociar la política al certificado de dispositivo.
Reemplazar`certificateArn`con`certificateArn`valor que ha guardado anteriormente en esta sección.

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

Si se ejecuta correctamente, este comando no devuelve nada.

6. Ejecute este comando para adjuntar el certificado de dispositivo alAWS IoTRecurso de cosa.
Reemplazar`certificateArn`con`certificateArn`valor que ha guardado anteriormente en esta sección.

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

Si se ejecuta correctamente, este comando no devuelve nada.

Después de aprovisionar correctamente el dispositivo enAWS IoT, ya está listo para seguir[the section called “Configuración delAWS IoTArchivo de configuración de Device Client y cliente de prueba MQTT para demostrar la comunicación MQTT”](#) (p. 150).

Configuración delAWS IoTArchivo de configuración de Device Client y cliente de prueba MQTT para demostrar la comunicación MQTT

Este procedimiento crea un archivo de configuración para probar elAWS IoTCliente del dispositivo.

Para crear el archivo de configuración para probar elAWS IoTCliente del dispositivo

1. En la ventana de terminal del equipo host local que está conectado a la Raspberry Pi:
 - a. Abra un editor de texto, comonano.
 - b. Copie este documento JSON y péguelo en el editor de texto abierto.

```
{  
    "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,  
        "log-level": "INFO"  
    }  
}
```

```
        "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. Reemplace el*punto de conexión*valor con endpoint de datos de dispositivo para suCuenta de AWSque encontraste enthe section called “Aprovisionamiento del dispositivo enAWS IoT Core” (p. 140).
- d. Guarde el archivo en el editor de texto como~/dc-configs/dc-pubsub-config.json.
- e. Ejecute este comando para establecer los permisos en el nuevo archivo de configuración.

```
chmod 644 ~/dc-configs/dc-pubsub-config.json
```

2. Para preparar elCliente de prueba MQTTpara suscribirse a todos los mensajes MQTT:
 - a. En el equipo host local, en elAWS IoTconsola, eligeCliente de prueba MQTT.
 - b. En el navegadorSuscripción a un tema, enFiltro de temas, introduzca#(un signo de una libra) y elijaSuscribirse.
 - c. Debajo de lasSuscripcionesetiqueta, confirma que ves#(un signo de una libra).

Deje la ventana con elCliente de prueba MQTTabrir mientras continúas con este tutorial.

Después de guardar el archivo y configurar elCliente de prueba MQTT, ya está listo para seguirthe section called “Paso 2: Demostrar la publicación de mensajes con elAWS IoTCliente del dispositivo” (p. 151).

Paso 2: Demostrar la publicación de mensajes con elAWS IoTCliente del dispositivo

Los procedimientos de esta sección demuestran cómo laAWS IoTDevice Client puede enviar mensajes MQTT predeterminados y personalizados.

Estas declaraciones de política de la política que ha creado en el paso anterior para estos ejercicios dan permiso a Raspberry Pi para ejecutar estas acciones:

- **iot:Connect**

Da nombre al clientePubSubTestThing, su Raspberry Pi ejecutando elAWS IoTCliente de dispositivo, para conectarse.

```
{  
    "Effect": "Allow",  
    "Action": [  
        "iot:Connect"  
    ],  
    "Resource": [  
        "arn:aws:iot:us-west-2:57EXAMPLE833:client/PubSubTestThing"  
    ]  
}
```

- **iot:Publish**

Otorga permiso a Raspberry Pi para publicar mensajes con un tema MQTT detest/dc/pubtopic.

```
{  
    "Effect": "Allow",  
    "Action": [  
        "iot:Publish"  
    ],  
    "Resource": [  
        "arn:aws:iot:us-west-2:57EXAMPLE833:topic/test/dc/pubtopic"  
    ]  
}
```

La **iot:Publish** acción da permiso para publicar en los temas MQTT enumerados en la matriz de recursos. La contenido de esos mensajes no está controlado por la declaración de política.

Publicar el mensaje predeterminado mediante elAWS IoTCliente del dispositivo

En este procedimiento se ejecuta elAWS IoTDevice Client para que publique un único mensaje MQTT predeterminado que indica que el Cliente de prueba MQTT recibe y muestra.

Para enviar el mensaje MQTT predeterminado desde elAWS IoTCliente del dispositivo

1. Asegúrese de que tanto la ventana del terminal del equipo host local que esté conectado a Raspberry Pi como la ventana con el Cliente de prueba MQTT están visibles mientras realiza este procedimiento.
2. En una ventana de terminal, escriba estos comandos para ejecutar laAWS IoTCliente de dispositivo mediante el archivo de configuración creado en [the section called “Crear el archivo de configuración” \(p. 143\)](#).

```
cd ~/aws-iot-device-client/build  
./aws-iot-device-client --config-file ~/dc-configs/dc-pubsub-config.json
```

En una ventana de terminal, AWS IoTDevice Client muestra mensajes de información y cualquier error que se produzca cuando se ejecuta.

Si no aparece ningún error en la ventana de la terminal, consulte la Cliente de prueba MQTT.

3. En el navegador Cliente de prueba MQTT, en el Suscripciones ventana, vea la Hello World! mensaje enviado a latest/dc/pubtopic tema de mensaje.
4. Si el archivo de AWS IoTDevice Client no muestra errores y verá Hello World! enviado a latest/dc/pubtopic message en el Cliente de prueba MQTT, ha demostrado una conexión correcta.
5. En una ventana de terminal, escriba ^C (Ctrl-C) para detener elAWS IoTCliente del dispositivo.

Después de demostrar que elAWS IoTDevice Client publicó el mensaje MQTT predeterminado, puede continuar con la[the section called “Publicar un mensaje personalizado mediante elAWS IoTCliente del dispositivo.” \(p. 153\).](#)

Publicar un mensaje personalizado mediante elAWS IoTCliente del dispositivo.

Los procedimientos de esta sección crean un mensaje MQTT personalizado y luego ejecutan elAWS IoTDevice Client para que publique el mensaje MQTT personalizado una vez para elCliente de prueba MQTT para recibir y mostrar.

Cree un mensaje MQTT personalizado para elAWS IoTCliente del dispositivo

Realice estos pasos en la ventana del terminal del equipo host local que está conectado a Raspberry Pi.

Para crear un mensaje personalizado para elAWS IoTCliente de dispositivo para publicar

1. En una ventana de terminal, abra un editor de texto, comonano.
2. En el editor de texto, copie y pegue el siguiente documento JSON. Esta será la carga útil del mensaje MQTT que elAWS IoTDevice Client publica.

```
{  
    "temperature": 28,  
    "humidity": 80,  
    "barometer": 1013,  
    "wind": {  
        "velocity": 22,  
        "bearing": 255  
    }  
}
```

3. Guarde el contenido del editor de texto como~/**messages/sample-ws-message.json**.
4. Escriba el siguiente comando para establecer los permisos del archivo de mensaje que acaba de crear.

```
chmod 600 ~/messages/*
```

Para crear un archivo de configuración para elAWS IoTCliente de dispositivo que se utilizará para enviar el mensaje personalizado

1. En la ventana de borna, en un editor de texto comonano, abra el existenteAWS IoTArchivo de configuración de Cliente de dispositivos:**~/dc-configs/dc-pubsub-config.json**.
2. Editar elsamplestenga un aspecto similar al siguiente. No es necesario cambiar ninguna otra parte de este archivo.

```
"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. Guarde el contenido del editor de texto como**~/dc-configs/dc-pubsub-custom-config.json**.
4. Ejecute este comando para establecer los permisos en el nuevo archivo de configuración.

```
chmod 644 ~/dc-configs/dc-pubsub-custom-config.json
```

Publicar el mensaje MQTT personalizado mediante elAWS IoTCliente del dispositivo

Este cambio solo afecta a contenidos de la carga útil de mensajes MQTT, por lo que la política actual seguirá funcionando. Sin embargo, si lasTema MQTT(tal como lo define elpublish-topicValor en~/dc-configs/dc-pubsub-custom-config.json) se ha cambiado, eliot::Publish También tendría que modificarse para permitir que Raspberry Pi publique en el nuevo tema MQTT.

Para enviar el mensaje MQTT desde elAWS IoTCliente del dispositivo

1. Asegúrese de que tanto la ventana de terminal como la ventana con elCliente de prueba MQTTestán visibles mientras realiza este procedimiento. Además, asegúrese de que suCliente de prueba MQTTsigue suscrito al#filtro de temas. Si no es así, suscríbase al#filtro de tema de nuevo.
2. En una ventana de terminal, escriba estos comandos para ejecutar laAWS IoTCliente de dispositivo mediante el archivo de configuración creado en[the section called “Crear el archivo de configuración” \(p. 143\)](#).

```
cd ~/aws-iot-device-client/build
./aws-iot-device-client --config-file ~/dc-configs/dc-pubsub-custom-config.json
```

En una ventana de terminal,AWS IoTDevice Client muestra mensajes de información y cualquier error que se produzca cuando se ejecuta.

Si no se muestran errores en la ventana del terminal, revise el cliente de prueba MQTT.

3. En el navegadorCliente de prueba MQTT, en elSuscripciones, consulte la carga útil de mensajes personalizados enviada a latest/dc/pubtopic tema de mensaje.
4. Si el archivo deAWS IoTDevice Client no muestra errores y ve la carga útil de mensajes personalizados que publicó en eltest/dc/pubtopicmessage en elCliente de prueba MQTT, ha publicado correctamente un mensaje personalizado.
5. En una ventana de terminal, escriba^C(Ctrl-C) para detener elAWS IoTCliente del dispositivo.

Después de demostrar que elAWS IoTDevice Client publicó una carga útil de mensajes personalizados, puede continuar[the section called “Paso 3: Demostrar suscribirse a mensajes con elAWS IoTCliente del dispositivo” \(p. 154\)](#).

Paso 3: Demostrar suscribirse a mensajes con elAWS IoTCliente del dispositivo

En esta sección, demostrará dos tipos de suscripciones a los mensajes:

- Suscripción de un solo tema
- Suscripción a temas comodín

Estas declaraciones de política de la política creada para estos ejercicios otorgan permiso a Raspberry Pi para realizar estas acciones:

- **iot:Receive**

Otorga elAWS IoTPermiso de cliente de dispositivo para recibir temas MQTT que coinciden con los mencionados en elResourceobjeto.

```
{
  "Effect": "Allow",
  "Action": [
    "iot:Receive"
```

```
],
"Resource": [
    "arn:aws:iot:us-west-2:57EXAMPLE833:topic/test/dc/subtopic"
]
}
```

- **iot:Subscribe**

Otorga elAWS IoTPermiso de cliente de dispositivo para suscribirse a filtros de temas MQTT que coinciden con los mencionados en elResourceobjeto.

```
{
    "Effect": "Allow",
    "Action": [
        "iot:Subscribe"
    ],
    "Resource": [
        "arn:aws:iot:us-west-2:57EXAMPLE833:topicfilter/test/dc/subtopic"
    ]
}
```

Suscribirse a un solo tema de mensaje MQTT

Este procedimiento muestra cómo elAWS IoTDevice Client puede suscribirse y registrar mensajes MQTT.

En la ventana de terminal del equipo host local que está conectado a la Raspberry Pi, enumere el contenido de `~/dc-configs/dc-pubsub-custom-config.json` abra el archivo en un editor de texto para revisar su contenido. Busque el complemento samples objeto, que debería tener un aspecto similar 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"
    }
}
```

Tenga en cuenta que `subscribe-topic` es el tema de MQTT al que elAWS IoTDevice Client se suscribirá cuando se ejecute. LaAWS IoTDevice Client escribe las cargas útiles de mensajes que recibe de esta suscripción en el archivo denominado en `subscribe-file`.

Para suscribirse a un tema de mensaje MQTT desde elAWS IoTCliente del dispositivo

1. Asegúrese de que tanto la ventana de terminal como la ventana con el cliente de prueba MQTT estén visibles mientras realiza este procedimiento. Además, asegúrese de que suCliente de prueba MQTTsigue suscrito al#filtro de temas. Si no es así, suscríbase al#filtro de tema de nuevo.
2. En una ventana de terminal, escriba estos comandos para ejecutar laAWS IoTCliente de dispositivo mediante el archivo de configuración creado en [the section called “Crear el archivo de configuración” \(p. 143\)](#).

```
cd ~/aws-iot-device-client/build
./aws-iot-device-client --config-file ~/dc-configs/dc-pubsub-custom-config.json
```

En una ventana de terminal,AWS IoTDevice Client muestra mensajes de información y cualquier error que se produzca cuando se ejecuta.

Si no aparece ningún error en la ventana de terminal, continúe en laAWS IoTconsola de .

3. En el navegadorAWS IoTConsola de, en elCliente de prueba MQTT, elige elPublicación de un temapestaña.
4. EnNombre del tema, introduzca**test/dc/subtopic**
5. EnCarga de mensaje, revisa el contenido del mensaje.
6. ElegirPublicaciónpara publicar el mensaje MQTT.
7. En una ventana de terminal, tenga en cuenta lamensaje recibidoentrada desde elAWS IoTClient de dispositivo que tiene un aspecto similar a este.

```
2021-11-10T16:02:20.890Z [DEBUG] {samples/PubSubFeature.cpp}: Message received on
subscribe topic, size: 45 bytes
```

8. Después de ver elmensaje recibidoentrada que muestra el mensaje recibido, introduzca^c(Ctrl-C) para detener elAWS IoTClient del dispositivo.
9. Introduzca este comando para ver el final del archivo de registro de mensajes y ver el mensaje que publicó desde elCliente de prueba MQTT.

```
tail ~/.aws-iot-device-client/log/pubsub_rx_msgs.log
```

Al ver el mensaje en el archivo de registro, ha demostrado que elAWS IoTDevice Client recibió el mensaje que publicó desde el cliente de prueba MQTT.

Suscribirse a varios temas de mensajes MQTT utilizando caracteres comodín

Estos procedimientos demuestran cómo elAWS IoTDevice Client puede suscribirse y registrar mensajes MQTT utilizando caracteres comodín. Para ello, harás lo siguiente:

1. Actualice el filtro de temas que elAWS IoTDevice Client utiliza para suscribirse a temas de MQTT.
2. Actualice la política utilizada por el dispositivo para permitir las nuevas suscripciones.
3. Ejecute laAWS IoTDevice Client y publica mensajes desde la consola de pruebas MQTT.

Para crear un archivo de configuración para suscribirse a varios temas de mensajes MQTT mediante un filtro de temas MQTT comodín

1. En la ventana de terminal del equipo host local que está conectado a la Raspberry Pi, abra-/**dc-configs/dc-pubsub-custom-config.json**para editar y localizar elsamplesojeto.
2. En el editor de texto, busque elsamplesojeto y actualiza el**subscribe-topic**tenga un aspecto similar al siguiente.

```
"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"
    }
}
```

El nuevosubscribe-topicValue es un[Filtro de temas MQTT \(p. 99\)](#)con un comodín MQTT al final. Esto describe una suscripción a todos los temas de MQTT que empiezan por**test/dc/**. LaAWS IoTDevice Client escribe las cargas útiles de mensajes que recibe de esta suscripción en el archivo denominado **subscribe-file**.

3. Guarde el archivo de configuración modificado como-/**dc-configs/dc-pubsub-wild-config.json** salga del editor.

Para modificar la política utilizada por su Raspberry Pi para permitir suscribirse y recibir varios temas de mensajes MQTT

1. En la ventana de terminal del equipo host local que está conectado a Raspberry Pi, en su editor de texto favorito, abra `~/policies/pubsub_test_thing_policy.json` para editar y, a continuación, busque `iot::Subscribe` y `iot::Received` declaraciones de política en el archivo.
2. En el navegador `iot::Subscribe` directiva, actualice la cadena del objeto Resource para reemplazar `subtopic` con `*`, para que tenga el siguiente aspecto.

```
{  
    "Effect": "Allow",  
    "Action": [  
        "iot:Subscribe"  
    ],  
    "Resource": [  
        "arn:aws:iot:us-west-2:57EXAMPLE833:topicfilter/test/dc/*"  
    ]  
}
```

Note

La [Caracteres comodín del filtro de temas MQTT \(p. 99\)](#) son los + (signo más) y # (signo de libra). Solicitud de suscripción con un # final se suscribe a todos los temas que empiezan por la cadena que precede al # carácter (por ejemplo, `test/dc/` en este caso).

Sin embargo, el valor del recurso de la declaración de política que autoriza esta suscripción debe utilizar un * (un asterisco) en lugar del # (un signo de libra) en el filtro de temas ARN. Esto se debe a que el procesador de políticas utiliza un carácter comodín diferente al que utiliza MQTT.

Para obtener más información sobre el uso de caracteres comodín para temas y filtros de temas en las políticas, consulte [Políticas para clientes MQTT \(p. 351\)](#).

3. En el navegador `iot::Received` directiva, actualice la cadena del objeto Resource para reemplazar `subtopic` con `*`, para que tenga el siguiente aspecto.

```
{  
    "Effect": "Allow",  
    "Action": [  
        "iot:Receive"  
    ],  
    "Resource": [  
        "arn:aws:iot:us-west-2:57EXAMPLE833:topic/test/dc/*"  
    ]  
}
```

4. Guarde el documento de política actualizado como `~/policies/pubsub_wild_test_thing_policy.json` salga del editor.
5. Introduzca este comando para actualizar la política de este tutorial y utilizar las nuevas definiciones de recursos.

```
aws iot create-policy-version \  
--set-as-default \  
--policy-name "PubSubTestThingPolicy" \  
--policy-document "file://~/policies/pubsub_wild_test_thing_policy.json"
```

Si el comando se ejecuta correctamente, devuelve una respuesta similar a esta. Observe que `policyVersion` es 2, lo que indica que es la segunda versión de esta política.

Si ha actualizado correctamente la política, puede continuar con el siguiente procedimiento.

```
{  
    "policyArn": "arn:aws:iot:us-west-2:57EXAMPLE833:policy/PubSubTestThingPolicy",  
    "policyDocument": "{\n        \"Version\": \"2012-10-17\",  
        \"Statement\": [\n            {\n                \"Effect\": \"Allow\",  
                \"Action\": \"iot:Connect\",  
                \"Resource\": \"arn:aws:iot:us-west-2:57EXAMPLE833:client/PubSubTestThing\"  
            },  
            {\n                \"Effect\": \"Allow\",  
                \"Action\": \"iot:Publish\",  
                \"Resource\": \"arn:aws:iot:us-west-2:57EXAMPLE833:topic/test/dc/pubtopic\"  
            },  
            {\n                \"Effect\": \"Allow\",  
                \"Action\": \"iot:Subscribe\",  
                \"Resource\": \"arn:aws:iot:us-west-2:57EXAMPLE833:topicfilter/test/dc/*\"  
            },  
            {\n                \"Effect\": \"Allow\",  
                \"Action\": \"iot:Receive\",  
                \"Resource\": \"arn:aws:iot:us-west-2:57EXAMPLE833:topic/test/dc/*\"  
            }  
        ],  
        \"policyVersionId\": \"2\",  
        \"isDefaultVersion\": true  
    }  
}
```

Si aparece un error de que hay demasiadas versiones de directivas para guardar una nueva, introduzca este comando para enumerar las versiones actuales de la política. Revise la lista que devuelve este comando para encontrar una versión de directiva que pueda eliminar.

```
aws iot list-policy-versions --policy-name "PubSubTestThingPolicy"
```

Escriba este comando para eliminar una versión que ya no necesita. Tenga en cuenta que no puede eliminar la versión de la política predeterminada. La versión predeterminada de la política es la que tiene `isDefaultVersion` valor `true`.

```
aws iot delete-policy-version \  
--policy-name "PubSubTestThingPolicy" \  
--policy-version-id policyId
```

Tras eliminar una versión de directiva, vuelva a intentar este paso.

Con el archivo de configuración y la política actualizados, está listo para demostrar las suscripciones de comodín con elAWS IoTClient del dispositivo.

Para demostrar cómo elAWS IoTDevice Client se suscribe y recibe varios temas de mensajes MQTT

1. En el navegadorCliente de prueba MQTT, revisa las suscripciones. Si el archivo deCliente de prueba MQTTestá suscrito a la en el#filtro de temas, continúe con el paso siguiente. Si no, en elCliente de prueba MQTT, enSuscripción a un tema, enFiltro de temas, introduzca#(un carácter de signo de libra) y, a continuación, elijaSuscribirsepara suscribirse a él.
2. En la ventana de terminal del equipo host local que está conectado a la Raspberry Pi, introduzca estos comandos para iniciar elAWS IoTClient del dispositivo.

```
cd ~/aws-iot-device-client/build  
./aws-iot-device-client --config-file ~/dc-configs/dc-pubsub-wild-config.json
```

3. Mientras veía elAWS IoTSalida de Device Client en la ventana de terminal del equipo host local, vuelva alCliente de prueba MQTT. En el navegadorPublicación de un tema, enNombre del tema, introduzca**test/dc/subtopic** luego seleccionePublicación.
4. En la ventana del terminal, confirme que el mensaje se ha recibido buscando un mensaje como:

```
2021-11-10T16:34:20.101Z [DEBUG] {samples/PubSubFeature.cpp}: Message received on subscribe topic, size: 76 bytes
```

5. Mientras veía elAWS IoTSalida de Device Client en la ventana de terminal del equipo host local, vuelva alCliente de prueba MQTT. En el navegadorPublicación de un tema, enNombre del tema, introduzca**test/dc/subtopic2**y luego seleccionePublicación.
6. En la ventana del terminal, confirme que el mensaje se ha recibido buscando un mensaje como:

```
2021-11-10T16:34:32.078Z [DEBUG] {samples/PubSubFeature.cpp}: Message received on subscribe topic, size: 77 bytes
```

7. Después de ver los mensajes que confirman que se han recibido ambos mensajes, introduzca^C(Ctrl-C) para detener elAWS IoTClient del dispositivo.
8. Introduzca este comando para ver el final del archivo de registro de mensajes y ver el mensaje que publicó desde elCliente de prueba MQTT.

```
tail -n 20 ~/.aws-iot-device-client/log/pubsub_rx_msgs.log
```

Note

El archivo de registro de contiene solo las cargas útiles de los mensajes. Los temas de mensajes no se graban en el archivo de registro de mensajes recibido.

También podría aparecer el mensaje publicado por elAWS IoTClient de dispositivo en el registro recibido. Esto se debe a que el filtro de temas comodín incluye ese tema del mensaje y, a veces, el agente de mensajes puede procesar la solicitud de suscripción antes de enviar el mensaje publicado a los suscriptores.

Las entradas del archivo de registro demuestran que se han recibido los mensajes. Puede repetir este procedimiento utilizando otros nombres de temas. Todos los mensajes que tienen un nombre de tema que comienza por**test/dc/**deben recibirse y registrarse. Los mensajes con nombres de temas que empiezan por cualquier otro texto se ignoran.

Después de demostrar cómo elAWS IoTDevice Client puede publicar mensajes MQTT y suscribirse a ellos, continuar[Tutorial: Demostrar acciones remotas \(trabajos\) con elAWS IoTClient del dispositivo \(p. 159\)](#).

Tutorial: Demostrar acciones remotas (trabajos) con elAWS IoTClient del dispositivo

En estos tutoriales, configurará e implementará trabajos en Raspberry Pi para demostrar cómo puede enviar operaciones remotas a sus dispositivos IoT.

Para iniciar este tutorial:

- Haga que su equipo host local se configure un Raspberry Pi como se utiliza en[la sección anterior \(p. 146\)](#).
- Si no has completado el tutorial en la sección anterior, puedes probar este tutorial utilizando la Raspberry Pi con una tarjeta microSD que tiene la imagen que guardaste después de instalar elAWS IoTClient del dispositivo en[\(Opcional\) Guardar la imagen de la tarjeta microSD \(p. 139\)](#).
- Si ya has ejecutado esta demo antes, revisa[??? \(p. 171\)](#)para eliminar todoAWS IoTRecursos que creó en ejecuciones anteriores para evitar errores de recursos duplicados.

Este tutorial tarda aproximadamente 45 minutos en completarse.

Cuando termine de examinar este tema:

- Habrás demostrado diferentes formas en que tu dispositivo IoT puede utilizar elAWS IoT Corepara ejecutar operaciones remotas gestionadas porAWS IoT.

Equipo necesario:

- El entorno de pruebas y desarrollo local que probó[una sección anterior \(p. 137\)](#)
- La Raspberry Pi que probó en[una sección anterior \(p. 137\)](#)
- La tarjeta de memoria microSD de la Raspberry Pi que probó en[una sección anterior \(p. 137\)](#)

Procedimientos de este tutorial

- [Paso 1: Preparación de la Raspberry Pi para ejecutar trabajos \(p. 160\)](#)
- [Paso 2: Crear y ejecutar el trabajo enAWS IoT \(p. 166\)](#)

Paso 1: Preparación de la Raspberry Pi para ejecutar trabajos

En los procedimientos de esta sección se describe cómo preparar su Raspberry Pi para ejecutar trabajos mediante elAWS IoTClient del dispositivo.

Note

Estos procedimientos son específicos del dispositivo. Si desea realizar los procedimientos de esta sección con más de un dispositivo al mismo tiempo, cada dispositivo necesitará su propia política y un certificado y nombre de cosa únicos específicos del dispositivo. Para proporcionar a cada dispositivo sus recursos únicos, realice este procedimiento una vez para cada dispositivo mientras cambia los elementos específicos del dispositivo tal como se describe en los procedimientos.

Procedimientos de este tutorial

- [Aprovisiona tu Raspberry Pi para demostrar trabajos \(p. 160\)](#)
- [Configuración delAWS IoTDevice Client para ejecutar el agente de trabajos \(p. 164\)](#)

Aprovisiona tu Raspberry Pi para demostrar trabajos

Los procedimientos de esta sección aprovisionar su Raspberry Pi enAWS IoTcreandoAWS IoTrecursos y certificados de dispositivo para ello.

[Cree y descargue archivos de certificado de dispositivo para demostrarloAWS IoTjobs](#)

Este procedimiento crea los archivos de certificado de dispositivo para esta demostración.

Si está preparando más de un dispositivo, este procedimiento debe realizarse en cada dispositivo.

Para crear y descargar los archivos de certificado de dispositivo para su Raspberry Pi:

En la ventana de terminal del equipo host local que está conectado a la Raspberry Pi, introduzca estos comandos.

1. Introduzca el siguiente comando para crear los archivos de certificado de dispositivo para su 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"
```

El comando devuelve una respuesta similar a la siguiente. Guarde el*certificateArn* valor para un uso posterior.

```
{  
  "certificateArn": "arn:aws:iot:us-  
west-2:57EXAMPLE833:cert/76e7e4edb3e52f52334be2f387a06145b2aa4c7fc810f3aea2d92abc227d269",  
  "certificateId": "76e7e4edb3e52f5233EXAMPLE7a06145b2aa4c7fc810f3aea2d92abc227d269",  
  "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"  
  }  
}
```

- Introduzca los siguientes comandos para establecer los permisos en el directorio de certificados y sus archivos.

```
chmod 700 ~/certs/jobs  
chmod 644 ~/certs/jobs/*  
chmod 600 ~/certs/jobs/private.pem.key
```

- Ejecute este comando para revisar los permisos de los directorios y archivos de certificados.

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

El resultado del comando debe ser el mismo que el que ve aquí, excepto que las fechas y horas del archivo serán 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
```

Después de descargar los archivos de certificado del dispositivo en su Raspberry Pi, estará listo para continuar[the section called “Aprovisiona tu Raspberry Pi para demostrar trabajos” \(p. 160\)](#).

CrearAWS IoT Recursos para demostrarAWS IoT jobs

Cree elAWS IoT Recursos para este dispositivo.

Si está preparando más de un dispositivo, este procedimiento debe realizarse para cada dispositivo.

Para aprovisionar el dispositivo enAWS IoT:

En la ventana de terminal del equipo host local que está conectado a la Raspberry Pi:

- Escriba el siguiente comando para obtener la dirección del punto de enlace de datos del dispositivo paraCuenta de AWS.

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

El valor del punto de enlace no ha cambiado desde la última vez que ejecutó este comando. Ejecutar el comando de nuevo aquí facilita la búsqueda y la pega del valor del extremo de datos en el archivo de configuración utilizado en este tutorial.

Ladescribe-endpointdevuelve una respuesta similar a la siguiente. Anote el**endpointAddress**valor para un uso posterior.

```
{  
  "endpointAddress": "a3qjEXAMPLEffp-ats.iot.us-west-2.amazonaws.com"  
}
```

2. Reemplazar**Nombre de cosa único**con un nombre único para el dispositivo. Si desea realizar este tutorial con varios dispositivos, indique su propio nombre a cada dispositivo. Por ejemplo,**TestDevice01**,**TestDevice02**, y así sucesivamente.

Escriba este comando para crear un nuevoAWS IoT recurso de objeto de su Raspberry Pi.

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

Ya que unAWS IoT cosa recurso es una representación virtual de su dispositivo en la nube, podemos crear múltiples recursos enAWS IoT para utilizar para diferentes fines. Todas ellas pueden ser utilizadas por el mismo dispositivo IoT físico para representar diferentes aspectos del dispositivo.

Estos tutoriales solo utilizarán un recurso a la vez por dispositivo. De esta forma, en estos tutoriales, representan las diferentes demostraciones para que después de crear elAWS IoT recursos para una demostración, puede volver atrás y repetir las demostraciones utilizando los recursos que creó específicamente para cada uno.

Si las recetasAWS IoT se creó un recurso de objeto, el comando devuelve una respuesta similar a esta. Anote el**thingArn**valor para utilizarlo más adelante cuando cree el trabajo para ejecutarse en este dispositivo.

```
{  
  "thingName": "uniqueThingName",  
  "thingArn": "arn:aws:iot:us-west-2:57EXAMPLE833:thing/uniqueThingName",  
  "thingId": "8ea78707-32c3-4f8a-9232-14bEXAMPLEfd"  
}
```

3. En la ventana de borna:

- a. Abra un editor de texto, comonano.
- b. Copie este documento JSON y péguelo en el editor de texto abierto.

```
{  
  "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. En el editor, en elResourcessección de cada declaración de política, sustitúyase**us-west - 2:57 EJEMPLO 833**con suRegión de AWS, un carácter de dos puntos (:) y tus 12 dígitosCuenta de AWSnúmero.
- d. En el editor, en cada declaración de política, sustituya**Nombre de cosa único**con el nombre de la cosa que le diste este recurso.
- e. Guarde el archivo en el editor de texto como-**policies/jobs_test_thing_policy.json**.

Si está ejecutando este procedimiento para varios dispositivos, guarde el archivo en este nombre de archivo en cada dispositivo.

4. Reemplazar**Nombre de cosa único**con el nombre de la cosa del dispositivo y, a continuación, ejecute este comando para crear unAWS IoTdirectiva adaptada para ese dispositivo.

```
aws iot create-policy \
--policy-name "JobTestPolicyForuniqueThingName" \
--policy-document "file://-/policies/jobs_test_thing_policy.json"
```

Si se crea la directiva, el comando devolverá una respuesta similar a esta.

```
{  
    "policyName": "JobTestPolicyForuniqueThingName",  
    "policyArn": "arn:aws:iot:us-west-2:57EXAMPLE833:policy/  
JobTestPolicyForuniqueThingName",  
    "policyDocument": "{\n        \"Version\": \"2012-10-17\",  
        \"Statement\": [\n            {\n                \"Effect\": \"Allow\",  
                \"Action\": \"iot:Connect\",  
                \"Resource\": \"arn:aws:iot:us-west-2:57EXAMPLE833:client/PubSubTestThing\"  
            },  
            {\n                \"Effect\": \"Allow\",  
                \"Action\": \"iot:Publish\",  
                \"Resource\": \"arn:aws:iot:us-west-2:57EXAMPLE833:topic/test/dc/pubtopic\"  
            },  
            {\n                \"Effect\": \"Allow\",  
                \"Action\": \"iot:Subscribe\",  
                \"Resource\": \"arn:aws:iot:us-west-2:57EXAMPLE833:topicfilter/test/dc/subtopic\"  
            },  
            {\n                \"Effect\": \"Allow\",  
                \"Action\": \"iot:Receive\",  
                \"Resource\": \"arn:aws:iot:us-west-2:57EXAMPLE833:topic/test/dc/*\"  
            }  
        ]  
    },  
    "policyVersionId": "1"
```

5. Reemplazar*Nombre de cosa único* con el nombre de la cosa para el dispositivo *certificateArn* con *certificateArn* valor guardado anteriormente en esta sección para este dispositivo y, a continuación, ejecute este comando para adjuntar la directiva al certificado de dispositivo.

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

Si se ejecuta correctamente, este comando no devuelve nada.

6. Reemplazar*Nombre de cosa único* con el nombre de la cosa del dispositivo, reemplace*certificateArn* con *certificateArn* valor guardado anteriormente en esta sección y, a continuación, ejecute este comando para adjuntar el certificado de dispositivo a laAWS IoTRecurso de objeto

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

Si se ejecuta correctamente, este comando no devuelve nada.

Después de aprovisionar correctamente su Raspberry Pi, estará listo para repetir esta sección para otro Raspberry Pi en la prueba o, si se han aprovisionado todos los dispositivos, continúe the section called “Configuración delAWS IoTDevice Client para ejecutar el agente de trabajos” (p. 164).

Configuración delAWS IoTDevice Client para ejecutar el agente de trabajos

Este procedimiento crea un archivo de configuración para elAWS IoTDevice Client para ejecutar el agente de trabajos:

Nota: si está preparando más de un dispositivo, este procedimiento debe realizarse en cada dispositivo.

Para crear el archivo de configuración para probar elAWS IoTClient del dispositivo:

1. En la ventana de terminal del equipo host local que está conectado a la Raspberry Pi:
 - a. Abra un editor de texto, comonano.
 - b. Copie este documento JSON y péguelo en el editor de texto abierto.

```
{
```

```
"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. Reemplace el *punto de conexión* valor con valor de punto final de datos de dispositivo para su Cuenta de AWS que encontraste en the section called “Aprovisionamiento del dispositivo en AWS IoT Core” (p. 140).
 - d. Reemplazar *Nombre de cosa único* con el nombre de la cosa que usaste para este dispositivo.
 - e. Guarde el archivo en el editor de texto como **~/dc-configs/dc-jobs-config.json**.
2. Ejecute este comando para establecer los permisos de archivo del nuevo archivo de configuración.

```
chmod 644 ~/dc-configs/dc-jobs-config.json
```

No usarás el Cliente de prueba MQTT para esta prueba. Mientras que el dispositivo intercambiará mensajes MQTT relacionados con los trabajos con AWS IoT, los mensajes de progreso del trabajo solo se intercambian con el dispositivo que ejecuta el trabajo. Dado que los mensajes de progreso del trabajo

solo se intercambian con el dispositivo que ejecuta el trabajo, no puedes suscribirte a ellos desde otro dispositivo, como elAWS IoTconsola de .

Después de guardar el archivo de configuración, estará listo para continuarthe section called “Paso 2: Crear y ejecutar el trabajo enAWS IoT” (p. 166).

Paso 2: Crear y ejecutar el trabajo enAWS IoT

Los procedimientos de esta sección crean un documento de trabajo y unAWS IoTrecurso de trabajo. Después de crear el recurso de trabajo,AWS IoTenvía el documento de trabajo a los destinos de trabajo especificados en los que un agente de trabajos aplica el documento de trabajo al dispositivo o cliente.

Procedimientos de esta sección

- [Crear y almacenar el documento de trabajo del trabajo \(p. 166\)](#)
- [Ejecute un trabajo enAWS IoTpara un dispositivo IoT \(p. 167\)](#)

Crear y almacenar el documento de trabajo del trabajo

Este procedimiento crea un documento de trabajo sencillo para incluirlo en unAWS IoTrecurso de trabajo. Este documento de trabajo muestra «¡Hola mundo!» en el objetivo del trabajo.

Para crear y almacenar un documento de trabajo:

1. Seleccione el depósito de Amazon S3 en el que guardará el documento de trabajo. Si no dispone de un bucket de Amazon S3 existente que utilizar para esto, tendrá que crear uno. Para obtener información acerca de cómo crear buckets de Amazon S3, consulte los temas de[Introducción a Amazon S3](#).
2. Crear y guardar el documento de trabajo para este trabajo
 - a. En el equipo host local, abra un editor de texto.
 - b. Copia y pega este texto en el editor.

```
{  
    "operation": "echo",  
    "args": ["Hello world!"]  
}
```
3. Reemplace el`path_to_file`con el camino `ahello-world-job.json`, si no está en el directorio actual, sustituya`s3_bucket_name`con la ruta del bucket de Amazon S3 al bucket que seleccionó y, a continuación, ejecute este comando para colocar el documento de trabajo en el bucket de Amazon S3.

```
aws s3api put-object \  
--key hello-world-job.json \  
--body path_to_file/hello-world-job.json --bucket s3_bucket_name
```

La URL del documento de trabajo que identifica el documento de trabajo que ha almacenado en Amazon S3 se determina reemplazando el`s3_bucket_nameyAWS_region`en la siguiente dirección URL. Registre la URL resultante para utilizarla más adelante como`job_document_path`

```
https://s3_bucket_name.s3.AWS_Region.amazonaws.com/hello-world-job.json
```

Note

AWSseguridad impide que puedas abrir esta URL fuera de tuCuenta de AWS, por ejemplo, mediante un navegador. La URL la utiliza elAWS IoTmotor de trabajos, que tiene acceso al archivo, de forma predeterminada. En un entorno de producción, tendrá que asegurarse de que suAWS IoTlos servicios tienen permiso para acceder a los documentos de trabajo almacenados en Amazon S3.

Después de guardar la URL del documento de trabajo, continúe en[the section called “Ejecute un trabajo enAWS IoTpara un dispositivo IoT” \(p. 167\)](#).

Ejecute un trabajo enAWS IoTpara un dispositivo IoT

Los procedimientos de esta sección inician laAWS IoTDevice Client en su Raspberry Pi para ejecutar el agente de trabajos en el dispositivo y esperar a que se ejecuten los trabajos. También crea un recurso de trabajo enAWS IoT, que enviará el trabajo a su dispositivo IoT y se ejecutará en él.

Note

Este procedimiento ejecuta un trabajo en un solo dispositivo.

Para iniciar el agente de trabajos en su Raspberry Pi:

1. En la ventana de terminal del equipo host local que está conectado a la Raspberry Pi, ejecute este comando para iniciar laAWS IoTClient del dispositivo.

```
cd ~/aws-iot-device-client/build
./aws-iot-device-client --config-file ~/dc-configs/dc-jobs-config.json
```

2. En la ventana de terminal, confirme que elAWS IoTDevice Client y muestra estos mensajes

```
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. En la ventana de terminal, después de ver este mensaje, continúe con el siguiente procedimiento y cree el recurso de trabajo. Tenga en cuenta que podría no ser la última entrada de la lista.

```
2021-11-15T18:45:56.785Z [INFO] {JobsFeature.cpp}: No pending jobs are scheduled,  
waiting for the next incoming job
```

Para crear unAWS IoT Recurso de trabajo

1. En el equipo host local:
 - a. Reemplazar`job_document_url`con la dirección URL del documento de trabajo[the section called "Crear y almacenar el documento de trabajo del trabajo" \(p. 166\)](#).
 - b. Reemplazar`thing_arn`con el ARN del recurso de cosa que creó para su dispositivo y, a continuación, ejecute este comando.

```
aws iot create-job \  
--job-id hello-world-job-1 \  
--document-source "job_document_url" \  
--targets "thing_arn" \  
--target-selection SNAPSHOT
```

Si se ejecuta correctamente, el comando devolverá un resultado similar a este.

```
{  
    "jobArn": "arn:aws:iot:us-west-2:57EXAMPLE833:job/hello-world-job-1",  
    "jobId": "hello-world-job-1"  
}
```

2. En la ventana de terminal, debería ver la salida delAWS IoTDevice Client así.

```
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 EphemeralPromise 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 EphemeralPromise 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. Mientras que elAWS IoTDevice Client se está ejecutando y esperando un trabajo, puede enviar otro trabajo cambiando eljob-idvalor y vuelva a ejecutar elcreate-jobdel paso 1.

Cuando haya terminado de ejecutar trabajos, en la ventana de terminal, escriba^C(Control-C) para detener elAWS IoTCliente del dispositivo.

Tutorial: Limpieza después de la ejecución del sistema AWS IoTTutorial del cliente del dispositivo

Los procedimientos de este tutorial le guiarán a través de la eliminación de los archivos y recursos que creó mientras completa los tutoriales de esta ruta de aprendizaje.

Procedimientos de este tutorial

- [Paso 1: Limpiar los dispositivos después de crear demostraciones con elAWS IoTCliente del dispositivo \(p. 169\)](#)
- [Paso 2: Limpieza deCuenta de AWSdespués de crear demostraciones con elAWS IoTCliente del dispositivo \(p. 171\)](#)

Paso 1: Limpiar los dispositivos después de crear demostraciones con elAWS IoTCliente del dispositivo

En este tutorial se describen dos opciones para limpiar la tarjeta microSD después de crear las demostraciones en esta ruta de aprendizaje. Elija la opción que proporciona el nivel de seguridad que necesita.

Tenga en cuenta que limpiar la tarjeta microSD del dispositivo no elimina ningunaAWS IoTRecursos que creó. Para limpiar elAWS IoTRecursos después de limpiar la tarjeta microSD del dispositivo, debe revisar el tutorial sobre[the section called “Limpieza después de crear demostraciones con elAWS IoTCliente del dispositivo” \(p. 171\)](#).

Opción 1: Limpieza reescribiendo la tarjeta microSD

La forma más sencilla y completa de limpiar la tarjeta microSD después de completar los tutoriales de esta ruta de aprendizaje es sobrescribir la tarjeta microSD con un archivo de imagen guardado que creaste mientras preparas tu dispositivo por primera vez.

Este procedimiento utiliza el equipo host local para escribir una imagen de tarjeta microSD guardada en una tarjeta microSD.

Note

Si el dispositivo no utiliza un medio de almacenamiento extraíble para su sistema operativo, consulte el procedimiento correspondiente a ese dispositivo.

Para escribir una nueva imagen en la tarjeta microSD

1. En el equipo host local, busque la imagen de la tarjeta microSD guardada que desea escribir en la tarjeta microSD.
2. Inserte la tarjeta microSD en el equipo host local.
3. Con una herramienta de imagen de tarjetas SD, escriba el archivo de imagen seleccionado en la tarjeta microSD.
4. Despues de escribir la imagen del sistema operativo Raspberry Pi en la tarjeta microSD, expulsa la tarjeta microSD y quitala de forma segura del equipo host local.

La tarjeta microSD está lista para utilizarse.

Opción 2: Limpieza eliminando directorios de usuario

Para limpiar la tarjeta microSD después de completar los tutoriales sin volver a escribir la imagen de la tarjeta microSD, puede eliminar los directorios de usuario individualmente. Esto no es tan exhaustivo como reescribir la tarjeta microSD desde una imagen guardada porque no elimina ningún archivo del sistema que pudiera haberse instalado.

Si la eliminación de los directorios de usuario es lo suficientemente exhaustiva para sus necesidades, puede seguir este procedimiento.

Para eliminar los directorios de usuario de esta ruta de aprendizaje del dispositivo

1. Ejecute estos comandos para eliminar los directorios de usuario, subdirectorios y todos sus archivos creados en esta ruta de aprendizaje, en la ventana de terminal conectada al dispositivo.

Note

Después de eliminar estos directorios y archivos, no podrá ejecutar las demostraciones sin volver a completar los tutoriales.

```
rm -Rf ~/dc-configs
rm -Rf ~/policies
rm -Rf ~/messages
rm -Rf ~/certs
rm -Rf ~/.aws-iot-device-client
```

2. Ejecute estos comandos para eliminar los directorios y archivos de origen de la aplicación, en la ventana del terminal conectada al dispositivo.

Note

Estos comandos no desinstalan ningún programa. Solo eliminan los archivos fuente utilizados para compilarlos e instalarlos. Despues de borrar estos archivos, el AWS CLI y el AWS IoT Device Client del dispositivo podría no funcionar.

```
rm -Rf ~/aws-cli
rm -Rf ~/aws
rm -Rf ~/aws-iot-device-client
```

Paso 2: Limpieza deCuenta de AWSdespués de crear demostraciones con elAWS IoT Cliente del dispositivo

Estos procedimientos le ayudan a identificar y eliminar elAWS Recursos que creó al completar los tutoriales de esta ruta de aprendizaje.

Eliminar recursosAWS IoT Recursos

Este procedimiento le ayuda a identificar y eliminar elAWS IoT Recursos que creó al completar los tutoriales de esta ruta de aprendizaje.

AWS IoT Recursos creados en esta ruta de aprendizaje

Tutorial	Recurso de objetos	Recurso de políticas
the section called “Instalar y configurar elAWS IoT Cliente del dispositivo” (p. 137)	Cosa de DevClitest	Política de cosas de DevClitest
the section called “Demostrar la comunicación de mensajes MQTT con elAWS IoT Cliente del dispositivo” (p. 146)	Lo subtest de Pub	Política de cosas de subprueba de Pub
the section called “Demostrar acciones remotas (trabajos) con elAWS IoT Cliente del dispositivo” (p. 159)	User defined(podría haber más de uno)	User defined(podría haber más de uno)

Para eliminar elAWS IoT Recursos, siga este procedimiento para cada recurso de cosa que haya creado

1. Reemplazar **thing_name** con el nombre del recurso de cosa que desea eliminar y, a continuación, ejecute este comando para enumerar los certificados adjuntos al recurso de cosa desde el equipo host local.

```
aws iot list-thing-principals --thing-name thing_name
```

Este comando devuelve una respuesta como esta que enumera los certificados que se adjuntan a **thing_name**. En la mayoría de los casos, solo habrá un certificado en la lista.

```
{  
    "principals": [  
        "arn:aws:iot:us-  
west-2:57EXAMPLE833:cert/23853eea3cf0edc7f8a69c74abeafa27b2b52823cab5b3e156295e94b26ae8ac"  
    ]  
}
```

2. Para cada certificado enumerado por el comando anterior:

- a. Reemplazar `certificate_ID` con el ID de certificado del comando anterior. El identificador del certificado son los caracteres alfanuméricos que siguen `cert/` en el ARN devuelto por el comando anterior. A continuación, ejecute este comando para desactivar el certificado.

```
aws iot update-certificate --new-status INACTIVE --certificate-id certificate_ID
```

Si se ejecuta correctamente, este comando no devuelve nada.

- b. Reemplazar `certificate_ARN` con el ARN del certificado de la lista de certificados devueltos anteriormente y, a continuación, ejecute este comando para enumerar las directivas adjuntas a este certificado.

```
aws iot list-attached-policies --target certificate_ARN
```

Este comando devuelve una respuesta como esta que enumera las directivas adjuntas al certificado. En la mayoría de los casos, solo habrá una política en la lista.

```
{  
    "policies": [  
        {  
            "policyName": "DevCliTestThingPolicy",  
            "policyArn": "arn:aws:iot:us-west-2:57EXAMPLE833:policy/  
DevCliTestThingPolicy"  
        }  
    ]  
}
```

- c. Para cada política asociada al certificado:

- i. Reemplazar `policy_name` con `policyName` valor del comando anterior, reemplace `certificate_ARN` con el ARN del certificado y, a continuación, ejecute este comando para separar la directiva del certificado.

```
aws iot detach-policy --policy-name policy_name --target certificate_ARN
```

Si se ejecuta correctamente, este comando no devuelve nada.

- ii. Reemplazar `policy_name` con `policyName` value y, a continuación, ejecute este comando para ver si la directiva está asociada a más certificados.

```
aws iot list-targets-for-policy --policy-name policy_name
```

Si el comando devuelve una lista vacía como esta, la política no se adjunta a ningún certificado y continúa enumerando las versiones de la política. Si aún hay certificados adjuntos a la política, continúe con `detach-thing-principal`.

```
{  
    "targets": []  
}
```

- iii. Reemplazar `policy_name` con `policyName` value y, a continuación, ejecute este comando para comprobar si hay versiones de políticas. Para eliminar la política, solo debe tener una versión.

```
aws iot list-policy-versions --policy-name policy_name
```

Si la política tiene solo una versión, como en este ejemplo, puede pasar al delete-policy paso y elimine la política ahora.

```
{  
    "policyVersions": [  
        {  
            "versionId": "1",  
            "isDefaultVersion": true,  
            "createDate": "2021-11-18T01:02:46.778000+00:00"  
        }  
    ]  
}
```

Si la política tiene más de una versión, como en este ejemplo, las versiones de políticas con un isDefaultVersion valor de false debe eliminarse antes de que se pueda eliminar la política.

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

Si necesitas eliminar una versión de política, reemplaza `policy_name` con `policyName` valor, sustituir `version_ID` con `versionId` valor del comando anterior y, a continuación, ejecute este comando para eliminar una versión de directiva.

```
aws iot delete-policy-version --policy-name policy_name --policy-version-id version_ID
```

Si se ejecuta correctamente, este comando no devuelve nada.

Después de eliminar una versión de política, repita este paso hasta que la directiva tenga solo una versión de política.

- iv. Reemplazar `policy_name` con `policyName` value y, a continuación, ejecute este comando para eliminar la política.

```
aws iot delete-policy --policy-name policy_name
```

- d. Reemplazar `thing_name` con el nombre de la cosa, reemplaza `certificate_ARN` con el ARN del certificado y, a continuación, ejecute este comando para separar el certificado del recurso de cosa.

```
aws iot detach-thing-principal --thing-name thing_name --principal certificate_ARN
```

Si se ejecuta correctamente, este comando no devuelve nada.

- e. Reemplazar `certificate_ID` con el ID de certificado del comando anterior. El identificador del certificado son los caracteres alfanuméricos que siguen `cert/` en el ARN devuelto por el comando anterior. A continuación, ejecute este comando para eliminar el recurso del certificado.

```
aws iot delete-certificate --certificate-id certificate_ID
```

Si se ejecuta correctamente, este comando no devuelve nada.

3. Reemplazar `thing_name` con el nombre de la cosa y, a continuación, ejecute este comando para eliminarla.

```
aws iot delete-thing --thing-name thing_name
```

Si se ejecuta correctamente, este comando no devuelve nada.

Eliminar recursosAWSrecursos

Este procedimiento le ayuda a identificar y eliminar otros AWSrecursos que creó al completar los tutoriales de esta ruta de aprendizaje.

Otro AWSrecursos creados en esta ruta de aprendizaje

Tutorial	Tipo de recurso	Nombre o ID de recurso
the section called “Demostrar acciones remotas (trabajos) con el AWS IoT Cliente del dispositivo” (p. 159)	Objeto Amazon S3	hello-world-job.json
the section called “Demostrar acciones remotas (trabajos) con el AWS IoT Cliente del dispositivo” (p. 159)	AWS IoT Recursos de trabajo	User defined

Para eliminar el AWSrecursos creados en esta ruta de aprendizaje

1. Para eliminar los trabajos creados en esta ruta de aprendizaje

- a. Ejecute este comando para enumerar los trabajos de tu Cuenta de AWS.

```
aws iot list-jobs
```

El comando devuelve una lista de AWS IoT trabajos en su Cuenta de AWS y Región de AWS. Su aspecto es el siguiente.

```
{
  "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 trabajo que reconozcas de la lista como trabajo que has creado en esta ruta de aprendizaje, reemplaza **jobId** con jobID valor del trabajo que se va a eliminar y, a continuación, ejecute este comando para eliminar un AWS IoT trabajo.

```
aws iot delete-job --job-id jobId
```

Si el comando se ejecuta correctamente, no devuelve nada.

2. Para eliminar los documentos del trabajo que almacenó en un bucket de Amazon S3 en esta ruta de aprendizaje.
- a. Reemplazar **bucket** con el nombre del bucket que utilizó y, a continuación, ejecute este comando para enumerar los objetos del bucket de Amazon S3 que utilizó.

```
aws s3api list-objects --bucket bucket
```

El comando devuelve una lista de los objetos Amazon S3 del bucket que se ve así.

```
{  
    "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": "\"bca60d5380b88e1a70cc27d321cabab72\"",  
            "Size": 29,  
            "StorageClass": "STANDARD",  
        }  
    ]  
}
```

```
        "Owner": {
            "DisplayName": "EXAMPLE",
            "ID": "e9e3d6ec1EXAMPLEf5fb5e6bd0a2b6ed03884d1ed392a82ad011c144736a4ee"
        }
    ]
}
```

- b. Para cada objeto que reconozca de la lista como objeto creado en esta ruta de aprendizaje, reemplace **bucket** con el nombre del bucket y **key** con el valor clave del objeto que se va a eliminar y, a continuación, ejecute este comando para eliminar un objeto Amazon S3.

```
aws s3api delete-object --bucket bucket --key key
```

Si el comando se ejecuta correctamente, no devuelve nada.

Después de borrar todos los AWS recursos y objetos que creó al completar esta ruta de aprendizaje, puede empezar de nuevo y repetir los tutoriales.

Creación de soluciones con elAWS IoTSDKs de dispositivos

Los tutoriales de esta sección le ayudan a explicar los pasos para desarrollar una solución de IoT que se puede implementar en un entorno de producción mediante AWS IoT.

Estos tutoriales pueden tardar más tiempo en completarse que los de la sección sobre [the section called "Creación de demostraciones con elAWS IoTCliente del dispositivo" \(p. 124\)](#) porque utilizan elAWS IoTSDK de dispositivos y explique los conceptos que se están aplicando con más detalle para ayudarlo a crear soluciones seguras y fiables.

Comience a crear soluciones con elAWS IoTSDKs de dispositivos

Estos tutoriales te guían a través de diferentes AWS IoT escenarios de. En su caso, en los tutoriales se utiliza la AWS IoTSDKs de dispositivos.

Temas

- [Tutorial: Conexión de un dispositivo a AWS IoT Core mediante el uso de la AWS IoTSDK de dispositivos \(p. 177\)](#)
- [Crear AWS IoT reglas para enrutar datos de dispositivos a otros servicios \(p. 193\)](#)
- [Conservación del estado del dispositivo mientras el dispositivo está desconectado con Device Shadows \(p. 223\)](#)
- [Tutorial: Creación de un autorizador personalizado para AWS IoT Core \(p. 244\)](#)
- [Tutorial: Control de la humedad del suelo con AWS IoT Raspberry Pi \(p. 256\)](#)

Tutorial: Conexión de un dispositivo aAWS IoT Coremediante el uso de laAWS IoTSDK de dispositivos

En este tutorial se muestra cómo conectar un dispositivo aAWS IoT Corepara que pueda enviar y recibir datos desde y haciaAWS IoT. Después de completar este tutorial, el dispositivo se configurará para conectarse aAWS IoT Corey entenderás cómo se comunican los dispositivos conAWS IoT.

En este tutorial, podrá:

1. the section called “Prepare el dispositivo paraAWS IoT” (p. 177)
2. the section called “Revisar el protocolo MQTT” (p. 178)
3. the section called “Revisar la aplicación de ejemplo pubsub.py Device SDK” (p. 178)
4. the section called “Connect del dispositivo y comuníquese conAWS IoT Core” (p. 184)
5. the section called “Consulte los resultados” (p. 189)

Para completar este tutorial se necesita aproximadamente una hora.

Antes de comenzar este tutorial, asegúrese de que dispone de:

- Completed[Introducción a AWS IoT Core \(p. 17\)](#)

En la sección de ese tutorial donde debes[the section called “Configuración del dispositivo” \(p. 42\)](#), seleccione la[the section called “Connect un Raspberry Pi u otro dispositivo” \(p. 56\)](#)para su dispositivo y utilice las opciones de lenguaje de Python para configurar el dispositivo.

- Mantén abierta la ventana de terminal que usas en ese tutorial porque también la usarás en este tutorial.
- Un dispositivo que puede ejecutar el[AWS IoTDevice SDK v2 para Python](#).

En este tutorial se muestra cómo conectar un dispositivo aAWS IoT Coreutilizando ejemplos de código de Python, que requieren un dispositivo relativamente potente.

Si trabaja con dispositivos con restricciones de recursos, es posible que estos ejemplos de código no funcionen en ellos. En ese caso, puede que tenga más éxito[the section called “Mediante AWS IoT Device SDK para Embedded C” \(p. 190\)“Hello, World!”](#)

Prepare el dispositivo paraAWS IoT

En[Introducción a AWS IoT Core \(p. 17\)](#), preparaste tu dispositivo yAWSCuenta para que puedan comunicarse. En esta sección se revisan los aspectos de esa preparación que se aplican a cualquier conexión de dispositivo conAWS IoT Core.

Para que un dispositivo se conecteAWS IoT Core:

1. Debe tener unCuenta de AWS.

El procedimiento de[Configurar suCuenta de AWS \(p. 18\)](#)describe cómo crear unCuenta de AWSsi todavía no dispone de uno.

2. En esa cuenta, debe disponer de lo siguienteAWS IoTRecursosdefinida para el dispositivo en suCuenta de AWSy Región.

El procedimiento de[CrearAWS IoTRecursos \(p. 38\)](#)describe cómo crear estos recursos para el dispositivo en suCuenta de AWSy Región.

- UNAcertificado de dispositivo registrado conAWS IoT se activa para autenticar el dispositivo.

El certificado se crea con frecuencia y se adjunta a unAWS IoT objeto de cosa. Si bien un objeto de cosa no es necesario para que un dispositivo se conecteAWS IoT, hace adicionalAWS IoT funciones disponibles para el dispositivo.

- UNApolítica de deadjunto al certificado de dispositivo que lo autoriza a conectarse aAWS IoT Core y realiza todas las acciones que quieras.

3. Un conexión a Internet que pueden acceder a suCuenta de AWS endpoints del dispositivo.

Los puntos finales del dispositivo se describen en[AWS IoT datos de dispositivos y puntos finales de servicio \(p. 78\)](#)y se puede ver en el[página de configuración delAWS IoT consola](#).

4. Software de comunicación tales como elAWS IoT proporcionan SDK de dispositivos. En este tutorial se utiliza[AWS IoT Device SDK v2 para Python](#).

Revisar el protocolo MQTT

Antes de hablar de la aplicación de muestra, es útil comprender el protocolo MQTT. El protocolo MQTT ofrece algunas ventajas sobre otros protocolos de comunicación de red, como HTTP, lo que lo convierte en una opción popular para dispositivos IoT. En esta sección se revisan los aspectos clave de MQTT que se aplican a este tutorial. Para obtener información acerca de cómo se compara MQTT con HTTP, consulte[Selección de un protocolo para la comunicación de su dispositivo \(p. 83\)](#).

MQTT utiliza un modelo de comunicación de publicación/suscripción

El protocolo MQTT utiliza un modelo de comunicación de publicación/suscripción con su host. Este modelo se diferencia del modelo solicitud/respuesta que utiliza HTTP. Con MQTT, los dispositivos establecen una sesión con el host que se identifica mediante un ID de cliente único. Para enviar datos, los dispositivos publican mensajes identificados por temas a un agente de mensajes del host. Para recibir mensajes del agente de mensajes, los dispositivos se suscriben a temas enviando filtros de temas en las solicitudes de suscripción al agente de mensajes.

MQTT admite sesiones persistentes

El agente de mensajes recibe mensajes de dispositivos y publica mensajes en dispositivos que se han suscrito a ellos. [sesiones persistentes \(p. 86\)](#)—sesiones que permanecen activas incluso cuando se desconecta el dispositivo iniciador; los dispositivos pueden recuperar mensajes publicados mientras se desconectaron. En el lado del dispositivo, MQTT admite niveles de calidad de servicio ([QoS \(p. 86\)](#)) que garantizan que el host recibe los mensajes enviados por el dispositivo.

Revisar la aplicación de ejemplo pubsub.py Device SDK

En esta sección se revisa el[pubsub.py](#) aplicación de ejemplo delAWS IoTDevice SDK v2 para Python que se utiliza en este tutorial. A continuación, revisaremos cómo se conecta aAWS IoT Core para publicar mensajes MQTT y suscribirse a ellos. En la siguiente sección se presentan algunos ejercicios para ayudarle a explorar cómo se conecta y se comunica un dispositivo conAWS IoT Core.

La[pubsub.py](#) aplicación de muestra muestra estos aspectos de una conexión MQTT conAWS IoT Core:

- [Protocolos de comunicación \(p. 179\)](#)
- [Sesiones persistentes \(p. 181\)](#)
- [Calidad del servicio \(p. 182\)](#)
- [Publicación de mensajes \(p. 182\)](#)
- [Suscripción a mensajes \(p. 183\)](#)
- [Desconexión y reconexión de dispositivos \(p. 184\)](#)

Protocolos de comunicación

La `pubsub.py` muestra una conexión MQTT utilizando los protocolos MQTT y MQTT a través de WSS. La [AWS Tiempo de ejecución común \(AWSCRT\)](#) proporciona soporte para protocolos de comunicación de bajo nivel y se incluye en el AWS IoT Device SDK v2 para Python.

MQTT

La `pubsub.py` muestra ejemplos de llamadas `mtls_from_path` (se muestra aquí) en `amqtt_connection_builder` establecer una conexión con AWS IoT Core mediante el protocolo MQTT. `mtls_from_path` utiliza certificados X.509 y TLS v1.2 para autenticar el dispositivo. La AWS Library for C++ gestiona los detalles de nivel inferior de esa conexión.

```
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`

Sus cuenta de AWS endpoint del dispositivo IoT

En la aplicación de ejemplo, este valor se transfiere desde la línea de comandos.

`cert_filepath`

La ruta al archivo de certificado del dispositivo

En la aplicación de ejemplo, este valor se transfiere desde la línea de comandos.

`pri_key_filepath`

Ruta del archivo de clave privada del dispositivo que se creó con su archivo de certificado

En la aplicación de ejemplo, este valor se transfiere desde la línea de comandos.

`ca_filepath`

La ruta al archivo de CA raíz de. Obligatorio solo si el servidor MQTT utiliza un certificado que aún no está en el almacén de confianza.

En la aplicación de ejemplo, este valor se transfiere desde la línea de comandos.

`client_bootstrap`

El objeto de tiempo de ejecución común que gestiona las actividades de comunicación de socket

En la aplicación de ejemplo, este objeto se crea una instancia antes de la llamada `amqtt_connection_builder.mtls_from_path`.

`on_connection_interrupted`, `on_connection_resumed`

Las funciones de devolución de llamada a llamar cuando se interrumpe y se reanuda la conexión del dispositivo

`client_id`

El identificador que identifica este dispositivo de forma exclusiva en laRegión de AWS

En la aplicación de ejemplo, este valor se transfiere desde la línea de comandos.

`clean_session`

Si desea iniciar una nueva sesión persistente o, si hay alguna, volver a conectarse a una ya existente

`keep_alive_secs`

El valor de mantener vivo, en segundos, para enviar elCONNECTrequest. Se enviará automáticamente un ping en este intervalo. Si el servidor no recibe un ping después de 1,5 veces este valor, asume que la conexión se ha perdido.

MQTT sobre WSS

La `pubsub.py` Ejemplos de llamadas `websockets_with_default_aws_signing` (se muestra aquí) en `amqtt_connection_builder` establecer una conexión con AWS IoT Core mediante el protocolo MQTT a través de WSS. `websockets_with_default_aws_signing` crea una conexión MQTT a través de WSS mediante `Signature V4` para autenticar el 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`

SusCuenta de AWS endpoint del dispositivo IoT

En la aplicación de ejemplo, este valor se transfiere desde la línea de comandos.

`client_bootstrap`

El objeto de tiempo de ejecución común que gestiona las actividades de comunicación de socket

En la aplicación de ejemplo, este objeto se crea una instancia antes de la llamada `amqtt_connection_builder.websockets_with_default_aws_signing`.

`region`

La AWS región de firma utilizada por la autenticación Signature V4. En `pubsub.py`, pasa el parámetro introducido en la línea de comandos.

En la aplicación de ejemplo, este valor se transfiere desde la línea de comandos.

`credentials_provider`

La AWS credenciales proporcionadas para utilizarlas para la autenticación

En la aplicación de ejemplo, este objeto se crea una instancia antes de la llamada `amqtt_connection_builder.websockets_with_default_aws_signing`.

`websocket_proxy_options`

Opciones de proxy HTTP, si se utiliza un host proxy

En la aplicación de ejemplo, este valor se inicializa antes de la llamada `amqtt_connection_builder.websockets_with_default_aws_signing`.

`ca_filepath`

La ruta al archivo de CA raíz de. Obligatorio solo si el servidor MQTT utiliza un certificado que aún no está en el almacén de confianza.

En la aplicación de ejemplo, este valor se transfiere desde la línea de comandos.

`on_connection_interrupted, on_connection_resumed`

Las funciones de devolución de llamada a llamar cuando se interrumpe y se reanuda la conexión del dispositivo

`client_id`

El identificador que identifica este dispositivo de forma exclusiva en laRegión de AWS.

En la aplicación de ejemplo, este valor se transfiere desde la línea de comandos.

`clean_session`

Si desea iniciar una nueva sesión persistente o, si hay alguna, volver a conectarse a una ya existente

`keep_alive_secs`

El valor de mantener vivo, en segundos, para enviar elCONNECTrequest. Se enviará automáticamente un ping en este intervalo. Si el servidor no recibe un ping después de 1,5 veces este valor, asume que la conexión se ha perdido.

HTTPS

¿Qué pasa con HTTPS?AWS IoT Core admite dispositivos que publican solicitudes HTTPS. Desde el punto de vista de la programación, los dispositivos envían solicitudes HTTPS aAWS IoT Core igual que cualquier otra aplicación. Para ver un ejemplo de un programa Python que envía un mensaje HTTP desde un dispositivo, consulte la[Ejemplo de código HTTPS \(p. 96\)](#)utilizando Pythonrequests. Este ejemplo envía un mensaje aAWS IoT Core usando HTTPS de tal manera queAWS IoT Core lo interpreta como un mensaje MQTT.

Si bienAWS IoT Core admite solicitudes HTTPS desde dispositivos, asegúrese de revisar la información sobre[Selección de un protocolo para la comunicación de su dispositivo \(p. 83\)](#)para que pueda tomar una decisión informada sobre qué protocolo utilizar para las comunicaciones de su dispositivo.

Sesiones persistentes

En la aplicación de ejemplo, configura la`clean_session`parámetro aFalseindica que la conexión debe ser persistente. En la práctica, esto significa que la conexión abierta por esta llamada se vuelve a conectar a una sesión persistente existente, si existe. De lo contrario, crea y se conecta a una nueva sesión persistente.

Con una sesión persistente, el agente de mensajes almacena los mensajes que se envían al dispositivo mientras el dispositivo no está conectado. Cuando un dispositivo se vuelve a conectar a una sesión persistente, el agente de mensajes envía al dispositivo cualquier mensaje almacenado al que se haya suscrito.

Sin una sesión persistente, el dispositivo no recibirá los mensajes que se envían mientras el dispositivo no está conectado. La opción que se debe utilizar depende de la aplicación y de si los mensajes que se

producen mientras un dispositivo no está conectado deben comunicarse. Para obtener más información, consulte [Uso de sesiones persistentes de MQTT \(p. 86\)](#).

Calidad del servicio

Cuando el dispositivo publica y se suscribe a los mensajes, se puede configurar la calidad de servicio (QoS) preferida. AWS IoT admite los niveles de QoS 0 y 1 para las operaciones de publicación y suscripción. Para obtener más información acerca de los niveles de QoS en AWS IoT, consulte [Opciones de calidad de servicio \(QoS\) de MQTT \(p. 86\)](#).

La AWS IoT runtime for Python define estas constantes para los niveles de QoS que admite:

Niveles de calidad de servicio de Python

Nivel de QoS MQTT	Valor simbólico de Python utilizado por SDK	Descripción
QoS nivel 0	<code>mqtt.QoS.AT_MOST_ONCE</code>	Solo se intentará enviar el mensaje, tanto si se recibe como si no. Es posible que el mensaje no se envíe en absoluto, por ejemplo, si el dispositivo no está conectado o se produce un error de red.
Nivel 1 de QoS de servicio	<code>mqtt.QoS.AT_LEAST_ONCE</code>	El mensaje se envía repetidamente hasta que <code>PUBACK</code> se recibe el acuse de recibo.

En la aplicación de ejemplo, las solicitudes de publicación y suscripción se realizan con un nivel de QoS de 1 (`mqtt.QoS.AT_LEAST_ONCE`).

- **QoS al publicar**

Cuando un dispositivo publica un mensaje con QoS nivel 1, envía el mensaje repetidamente hasta que recibe un `PUBLISH` respuesta del agente de mensajes. Si el dispositivo no está conectado, el mensaje se pone en cola para enviarlo después de volver a conectarse.

- **QoS al suscribirse**

Cuando un dispositivo se suscribe a un mensaje con QoS nivel 1, el agente de mensajes guarda los mensajes a los que está suscrito el dispositivo hasta que se puedan enviar al dispositivo. El agente de mensajes vuelve a enviar los mensajes hasta que recibe un `PUBLISH` respuesta desde el dispositivo.

Publicación de mensajes

Después de establecer correctamente una conexión a AWS IoT Core, los dispositivos pueden publicar mensajes. La `pubsub.py` muestra lo hace llamando al `publish` funcionamiento del `mqtt_connection` objeto.

```
mqtt_connection.publish(  
    topic=args.topic,  
    payload=message,  
    qos= mqtt.QoS.AT_LEAST_ONCE  
)
```

topic

El nombre del tema del mensaje que identifica el mensaje

En la aplicación de ejemplo, esto se pasa desde la línea de comandos.

payload

La carga útil del mensaje con formato de cadena (por ejemplo, un documento JSON)

En la aplicación de ejemplo, esto se pasa desde la línea de comandos.

Un documento JSON es un formato de carga útil común y uno reconocido por otros AWS IoT servicios; sin embargo, el formato de datos de la carga útil del mensaje puede ser cualquier cosa que los editores y suscriptores acuerden. Otros AWS IoT servicios, sin embargo, solo reconocen JSON y CBOR, en algunos casos, para la mayoría de las operaciones.

qos

El nivel de QoS de este mensaje

Suscripción a mensajes

Para recibir mensajes de AWS IoT otros servicios y dispositivos, los dispositivos se suscriben a esos mensajes por el nombre de su tema. Los dispositivos pueden suscribirse a mensajes individuales especificando un [nombre del tema \(p. 98\)](#) y a un grupo de mensajes especificando un [filtro de temas \(p. 99\)](#), que puede incluir caracteres comodín. La `pubsub.py` muestra el código que se muestra aquí para suscribirse a los mensajes y registrar las funciones de devolución de llamada para procesar el mensaje una vez recibido.

```
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

El tema al que se suscribe. Puede ser un nombre de tema o un filtro de temas.

En la aplicación de ejemplo, esto se pasa desde la línea de comandos.

qos

Si el agente de mensajes debe almacenar estos mensajes mientras el dispositivo está desconectado.

Un valor de `mqtt.QoS.AT LEAST_ONCE` (nivel de QoS 1), requiere que se especifique una sesión persistente (`clean_session=False`) cuando se crea la conexión.

callback

Función a la que se debe llamar para procesar el mensaje suscrito.

`mqtt_connection.subscribe` devuelve un futuro y un ID de paquete. Si la solicitud de suscripción se inició correctamente, el ID de paquete devuelto es mayor que 0. Para asegurarse de que la suscripción ha sido recibida y registrada por el agente de mensajes, debe esperar a que devuelva el resultado de la operación asíncrona, como se muestra en el ejemplo de código.

La función de devolución de llamada

La devolución de llamada en el `pubsub.py` procesa los mensajes suscritos a medida que el dispositivo los recibe.

```
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

El tema del mensaje

Este es el nombre del tema específico del mensaje recibido, incluso si se ha suscrito a un filtro de temas.

payload

La carga del mensaje

El formato para esto es específico de la aplicación.

kwargs

Posibles argumentos adicionales, tal como se describe en [MQTT.Connection.subscribe](#).

En el `pubsub.py` muestra, `on_message_received` solo muestra el tema y su carga útil. También cuenta los mensajes recibidos para finalizar el programa una vez alcanzado el límite.

La aplicación evaluaría el tema y la carga útil para determinar qué acciones debe realizar.

Desconexión y reconexión de dispositivos

El `pubsub.py` incluye funciones de devolución de llamada a las que se llama cuando se desconecta el dispositivo y cuando se restablece la conexión. Las acciones que realiza su dispositivo en estos eventos son específicas de la aplicación.

Cuando un dispositivo se conecta por primera vez, debe suscribirse a los temas para recibir. Si la sesión de un dispositivo está presente cuando se vuelve a conectar, se restauran sus suscripciones y los mensajes almacenados de esas suscripciones se envían al dispositivo después de volver a conectarse.

Si la sesión de un dispositivo ya no existe cuando se vuelve a conectar, debe volver a suscribirse a sus suscripciones. Las sesiones persistentes tienen una vida útil limitada y pueden caducar cuando el dispositivo se desconecta durante demasiado tiempo.

Conect del dispositivo y comuníquese con AWS IoT Core

En esta sección se presentan algunos ejercicios que le ayudarán a explorar diferentes aspectos de la conexión del dispositivo aAWS IoT Core. En estos ejercicios, utilizará la [Cliente de prueba MQTT](#) en la AWS IoT consola para ver qué publica el dispositivo y publicar mensajes en el dispositivo. Estos ejercicios utilizan el `pubsub.py` muestra de la [AWS IoT Device SDK v2 para Python](#) y aprovecha tu experiencia con [Introducción a AWS IoT Core \(p. 17\)](#).

En esta sección, harás lo siguiente:

- [Suscríbete a filtros de temas comodín \(p. 185\)](#)
- [Suscripciones de filtros de temas de proceso \(p. 186\)](#)

- [Publicar mensajes desde tu dispositivo \(p. 188\)](#)

Para estos ejercicios, comenzarás desde el `pubsub.py` programa de ejemplo.

Note

En estos ejercicios se supone que ha completado el [Introducción a AWS IoT Core \(p. 17\)](#) tutoriales y utilice la ventana de terminal de su dispositivo desde ese tutorial.

Suscríbete a filtros de temas comodín

En este ejercicio, modificará la línea de comandos utilizada para llamar `pubsub.py` para suscribirse a un filtro de temas comodín y procesar los mensajes recibidos en función del tema del mensaje.

Procedimiento de ejercicio

Para este ejercicio, imagina que tu dispositivo contiene un control de temperatura y un control de luz. Utiliza estos nombres de temas para identificar los mensajes sobre ellos.

1. Antes de iniciar el ejercicio, intente ejecutar este comando desde el [Introducción a AWS IoT Core \(p. 17\)](#) tutoriales en tu dispositivo para asegurarte de que todo esté listo para el ejercicio.

```
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
```

Debería ver la misma salida que vio en el [Explicación introductoria \(p. 60\)](#).

2. Para este ejercicio, cambie estos parámetros de línea de comandos.

Acción	Command line parameter	Efecto
agregar	--message ""	Configurar <code>pubsub.py</code> para escuchar solo
agregar	--count 2	Finalizar el programa después de recibir dos mensajes
cambiar	--topic device/+.details	Definir el filtro de temas al que desea suscribirse

La realización de estos cambios en la línea de comandos inicial da como resultado esta línea de comandos. Introduzca este comando en la ventana de terminal del 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
```

El programa debe mostrar algo similar al siguiente:

```
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...
```

Si ves algo así en tu terminal, tu dispositivo está listo y escucha los mensajes en los que empiezan los nombres de los temas `topic-1/y` termine con/`detail`. Así que, vamos a probar eso.

3. Aquí hay un par de mensajes que puede recibir el dispositivo.

Nombre del tema	Carga de mensaje
<code>device/temp/details</code>	{ "desiredTemp": 20, "currentTemp": 15 }
<code>device/light/details</code>	{ "desiredLight": 100, "currentLight": 50 }

4. mediante el cliente de prueba MQTT enAWS IoT, envía los mensajes descritos en el paso anterior al dispositivo.
 - a. Abra el icono [Cliente de prueba MQTT](#) en laAWS IoT consola de .
 - b. En Suscripción a un tema, en el Campo de temas de suscripción, escriba el filtro de temas:`device/+details` y luego seleccione Suscripción al tema.
 - c. En el navegador Suscripciones Columna del cliente de prueba MQTT, elija dispositivo/+detalles.
 - d. Para cada uno de los temas de la tabla anterior, haga lo siguiente en el cliente de prueba MQTT:
 1. En Publicación, introduzca el valor de Nombre del tema columna de la tabla.
 2. En el campo de carga útil del mensaje que se encuentra debajo del nombre del tema, escriba el valor de Carga de mensaje columna de la tabla.
 3. Observe la ventana de la terminal donde `pubsub.py` se está ejecutando y, en el cliente de prueba MQTT, elija Publicación de un tema.

Debería ver que el mensaje ha sido recibido por `pubsub.py` en la ventana de terminal.

Resultado del ejercicio

Con esto, `pubsub.py`, se suscribió a los mensajes mediante un filtro de temas comodín, los recibió y los ha mostrado en la ventana del terminal. Observe cómo se suscribió a un único filtro de temas y se ha llamado a la función de devolución de llamada para procesar mensajes que tienen dos temas distintos.

Suscripciones de filtros de temas de proceso

Basándose en el ejercicio anterior, modifique la `pubsub.py` aplicación de ejemplo para evaluar los temas de los mensajes y procesar los mensajes suscritos en función del tema.

Procedimiento de ejercicio

Para evaluar el tema del mensaje

1. Copie `pubsub.py` en `pubsub2.py`.
2. Abra `pubsub2.py` en el editor de texto o IDE que prefiera.
3. En `pubsub2.py`, encuentre la función `on_message_received`.
4. En `on_message_received`, inserte el siguiente código después de la línea que comienza `conprint("Received message")` antes de la línea que comienza `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. Guarde los cambios y ejecute el programa modificado mediante esta línea de comandos.

```
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. En el navegador AWS IoT, abra la[Cliente de prueba MQTT](#).
7. En Suscripción a un tema, en el Campo de temas de suscripción, escriba el filtro de temas:**device/+/details** luego seleccione Suscripción al tema.
8. En el navegador Suscripciones Columna del cliente de prueba MQTT, elija dispositivo/+/detalles.
9. Para cada uno de los temas de esta tabla, haga lo siguiente en el cliente de prueba MQTT:

Nombre del tema	Carga de mensaje
device/temp/details	{ "desiredTemp": 20, "currentTemp": 15 }
device/light/details	{ "desiredLight": 100, "currentLight": 50 }

1. En Publicación, introduzca el valor de Nombre del tema columna de la tabla.
2. En el campo de carga útil del mensaje que se encuentra debajo del nombre del tema, escriba el valor de Carga de mensaje columna de la tabla.
3. Observe la ventana de la terminal donde `pubsub.py` se está ejecutando y, en el cliente de prueba MQTT, elija Publicación de un tema.

Debería ver que el mensaje ha sido recibido por `pubsub.py` en la ventana de terminal.

Debería ver algo similar al siguiente en la ventana de su 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 del ejercicio

En este ejercicio, ha agregado código para que la aplicación de ejemplo reconozca y procese varios mensajes en la función de devolución de llamada. Con esto, tu dispositivo podría recibir mensajes y actuar en función de ellos.

Otra forma de que su dispositivo reciba y procese varios mensajes es suscribirse a diferentes mensajes por separado y asignar cada suscripción a su propia función de devolución de llamada.

Publicar mensajes desde tu dispositivo

Puede utilizar la aplicación de ejemplo pubsub.py para publicar mensajes desde su dispositivo. Aunque publicará los mensajes tal cual, los mensajes no se pueden leer como documentos JSON. Este ejercicio modifica la aplicación de ejemplo para poder publicar documentos JSON en la carga útil de mensajes que puede leer AWS IoT Core.

Procedimiento de ejercicio

En este ejercicio, se enviará el siguiente mensaje con el device/data tema.

```
{  
    "timestamp": 1601048303,  
    "sensorId": 28,  
    "sensorData": [  
        {  
            "sensorName": "Wind speed",  
            "sensorValue": 34.2211224  
        }  
    ]  
}
```

Para preparar al cliente de prueba MQTT para supervisar los mensajes de este ejercicio

1. En Suscripción a un tema, en el Campo de temas de suscripción, escriba el filtro de temas:**device/data** y luego seleccione Suscripción al tema.
2. En el navegador Suscripciones Columna del cliente de prueba MQTT, elija dispositivo/datos.
3. Mantén abierta la ventana del cliente de prueba de MQTT para esperar los mensajes de tu dispositivo.

Para enviar documentos JSON con la aplicación de ejemplo pubsub.py

1. En tu dispositivo, copia `pubsub.py` a `pubsub3.py`.
2. Edita `pubsub3.py` para cambiar el formato de los mensajes que publica.
 - a. Abra `pubsub3.py` en un editor de texto.
 - b. Busque esta línea de código:

```
message = "{} [{}]".format(args.message, publish_count)
```
 - c. Cámbialo a:

```
message = "{}".format(args.message)
```
 - d. Guarde los cambios.
3. En el dispositivo, ejecuta este comando para enviar el mensaje dos veces.

```
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
```

- En el cliente de prueba MQTT, compruebe que ha interpretado y formateado el documento JSON en la carga útil del mensaje, como este:

The screenshot shows the AWS IoT Device Data interface. At the top, it says "device/data" and "September 25, 2020, 08:57:14 (UTC-0700)". On the right, there are "Export" and "Hide" buttons. The main area displays a JSON message:

```
{  
  "timestamp": 1601048303,  
  "sensorId": 28,  
  "sensorData": [  
    {  
      "sensorName": "Wind speed",  
      "sensorValue": 34.2211224  
    }  
  ]  
}
```

Por defecto, `pubsub3.py` también se suscribe a los mensajes que envía. Debería ver que ha recibido los mensajes en la salida de la aplicación. La ventana de la terminal tendrá un aspecto similar al siguiente.

```
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 del ejercicio

Con esto, tu dispositivo puede generar mensajes para enviarlos aAWS IoT Core para probar la conectividad básica y proporcionar mensajes de dispositivo paraAWS IoT Core procesar. Por ejemplo, puedes usar esta aplicación para enviar datos de prueba desde tu dispositivo a pruebaAWS IoT acciones de reglas.

Consulte los resultados

Los ejemplos de este tutorial le brindan experiencia práctica con los conceptos básicos de cómo los dispositivos pueden comunicarse conAWS IoT Core—una parte fundamental de suAWS IoT solución. Cuando tus dispositivos pueden comunicarse conAWS IoT Core, pueden pasar mensajes aAWS servicios y otros dispositivos en los que pueden actuar. Del mismo modo,AWS servicios y otros dispositivos pueden procesar información que da como resultado mensajes enviados de vuelta a sus dispositivos.

Cuando esté preparado para explorarAWS IoT CoreAdemás, prueba estos tutoriales:

- the section called “Envío de una notificación de Amazon SNS” (p. 201)
- the section called “Almacenamiento de datos de dispositivos en una tabla de DynamoDB” (p. 209)
- the section called “Dar formato a una notificación mediante unAWS Lambda función” (p. 214)

Tutorial: Mediante AWS IoT Device SDK para Embedded C

En esta sección, se explica cómo se ejecuta el AWS IoT Device SDK para Embedded C.

Procedimientos de esta sección

- [Paso 1: Instalar la AWS IoT Device SDK para Embedded C \(p. 190\)](#)
- [Paso 2: Configurar la aplicación de muestra \(p. 190\)](#)
- [Paso 3: Crear y ejecutar la aplicación de ejemplo \(p. 192\)](#)

Paso 1: Instalar la AWS IoT Device SDK para Embedded C

AWS IoT Device SDK para Embedded C generalmente se dirige a dispositivos con limitaciones de recursos que requieren un tiempo de ejecución optimizado del lenguaje C. Puede usar el SDK en cualquier sistema operativo y alojarlo en cualquier tipo de procesador (por ejemplo, MCU y MPU). Si dispone de más memoria y recursos de procesamiento, le recomendamos que utilice uno de los pedidos superiores.AWS IoT SDK para dispositivos y dispositivos móviles (por ejemplo, C++, Java, JavaScript Python).

En general, AWS IoT Device SDK para Embedded C está dirigido a sistemas que utilizan MCU o MPU de bajo rendimiento que ejecutan sistemas operativos integrados. Para el ejemplo de programación de esta sección, asumimos que su dispositivo utiliza Linux.

Example

1. Descarga deAWS IoT Device SDK para Embedded Ca tu dispositivo desde[GitHub](#).

```
git clone https://github.com/aws/aws-iot-device-sdk-embedded-c.git --recurse-submodules
```

Esto crea un directorio denominado aws-iot-device-sdk-embedded-c en el directorio actual.

2. Vaya a ese directorio y retire la última versión. Consulte[github.com/aws/aws-iot-device-SDK incrustadas-C/etiquetas](#)Para obtener la etiqueta de versión más reciente.

```
cd aws-iot-device-sdk-embedded-c
git checkout latest-release-tag
```

3. Instale OpenSSL versión 1.1.0 o posterior. Las bibliotecas de desarrollo de OpenSSL suelen denominarse «libssl-dev» o «openssl-devel» cuando se instalan a través de un gestor de paquetes.

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

Paso 2: Configurar la aplicación de muestra

El AWS IoT Device SDK para Embedded C contiene aplicaciones de ejemplo que puede probar. Para simplificar el proceso, este tutorial utiliza `lambda_demo_mutual_auth`, que ilustra cómo conectarse a laAWS IoT Coreagente de mensajes y suscribirse y publicar en temas MQTT.

1. Copie el certificado y la clave privada que creó en[Introducción a AWS IoT Core \(p. 17\)](#)en elbuild/bin/certificatesdirectorio.

Note

Los certificados de dispositivo y de entidad de certificación raíz están sujetos a vencimiento o revocación. Si estos certificados caducan o se revocan, debe copiar un nuevo certificado de CA o un certificado de clave privada y dispositivo en el dispositivo.

2. Debe configurar el ejemplo con su punto de enlace de AWS IoT Core, su clave privada, su certificado y su certificado de la entidad de certificación raíz personales. Vaya al directorio `aws-iot-device-sdk-embedded-c/demos/mqtt/mqtt_demo_mutual_auth`.

Si tiene laAWS CLI instalada, puede utilizar este comando para buscar la URL del punto de enlace de su cuenta.

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

Si no tiene instalada la AWS CLI, abra la [consola de AWS IoT](#). Desde el panel de navegación, elija Manage (Administrar) y, a continuación, Things (Objetos). Elija la cosa IoT de su dispositivo y, a continuación, elija Interactuar. Su punto de enlace se muestra en la sección HTTPS de la página de detalles del objeto.

3. Abra el icono `demo_config.h` y actualice los valores de los elementos siguientes:

`AWS_IOT_ENDPOINT`

Su punto de enlace personal.

`CLIENT_CERT_PATH`

La ruta del archivo de certificado, por ejemplo `certificates/device.pem.crt`.

`CLIENT_PRIVATE_KEY_PATH`

Por ejemplo, el nombre del archivo de la clave privada `certificates/private.pem.key`.

Por ejemplo:

```
// 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. Compruebe si tiene CMake instalado en el dispositivo mediante este comando.

```
cmake --version
```

Si ve la información de la versión del compilador, puede continuar con la siguiente sección.

Si aparece un error o no ve la información, deberá instalar el paquete `cmake` mediante este comando.

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

Ejecute `lacmake --version` y confirme que CMake se ha instalado y que está listo para continuar.

5. Compruebe si tiene las herramientas de desarrollo instaladas en su dispositivo mediante este comando.

```
gcc --version
```

Si ve la información de la versión del compilador, puede continuar con la siguiente sección.

Si aparece un error o no ve la información del compilador, deberá instalar el paquete `build-essential` con este comando.

```
sudo apt-get install build-essential
```

Vuelva a ejecutar el comando `gcc --version` y confirme que las herramientas de compilación se han instalado y que está listo para continuar.

Paso 3: Crear y ejecutar la aplicación de ejemplo

Para ejecutar las aplicaciones de ejemplo de AWS IoT Device SDK para Embedded C

1. Vaya `aaws-iot-device-sdk-embedded-cy` crea un directorio de compilación.

```
mkdir build && cd build
```

2. Introduzca el siguiente comando CMake para generar los Makefiles necesarios para crear.

```
cmake ..
```

3. Escriba el siguiente comando para compilar el archivo de aplicación ejecutable.

```
make
```

4. Ejecute la aplicación `mqtt_demo_mutual_auth` con este comando.

```
cd bin  
./mqtt_demo_mutual_auth
```

Debería ver un resultado similar a este:

```
[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.
```

El dispositivo está conectado aAWS IoT utilizando elAWS IoT Device SDK para Embedded C.

También puede utilizar laAWS IoT para ver los mensajes MQTT que está publicando la aplicación de muestra. Para obtener información sobre cómo utilizar el cliente MQTT en la [consola de AWS IoT](#), consulte the section called “Ver los mensajes MQTT con el cliente MQTT de AWS IoT” (p. 65).

CrearAWS IoTreglas para enrutar datos de dispositivos a otros servicios

En estos tutoriales, se explica cómo crear y probarAWS IoTreglas que utilizan algunas de las acciones de reglas más comunes.

AWS IoTreglas envían datos desde tus dispositivos a otrosAWSServicios de . Escuchan mensajes MQTT específicos, dan formato a los datos de las cargas útiles de los mensajes y envían el resultado a otrosAWSServicios de .

Te recomendamos que los pruebes en el orden en que se muestran aquí, incluso si tu objetivo es crear una regla que utilice una función Lambda o algo más complejo. Los tutoriales se presentan en orden desde lo básico hasta lo complejo. Presentan nuevos conceptos de forma incremental para ayudarle a aprender los conceptos que puede utilizar para crear acciones de reglas que no tienen un tutorial específico.

Note

AWS IoTreglas le ayudan a enviar los datos de sus dispositivos IoT a otrosAWSServicios de . Sin embargo, para hacerlo correctamente, necesita un conocimiento práctico de los demás servicios a los que desea enviar datos. Si bien estos tutoriales proporcionan la información necesaria para completar las tareas, puede resultarle útil obtener más información sobre los servicios a los que desea enviar datos antes de utilizarlos en la solución. Una explicación detallada de la otraAWSlos servicios están fuera del ámbito de estos tutoriales.

Información general del escenario del tutorial

El escenario de estos tutoriales es el de un dispositivo sensor meteorológico que publica periódicamente sus datos. Hay muchos de estos dispositivos sensores en este sistema imaginario. Sin embargo, los

tutoriales de esta sección se centran en un solo dispositivo y muestran cómo puede acomodar varios sensores.

En los tutoriales de esta sección se muestra cómo utilizarAWS IoTreglas para realizar las siguientes tareas con este sistema imaginario de dispositivos sensores meteorológicos.

- [Tutorial: Republicación de un mensaje MQTT \(p. 195\)](#)

En este tutorial se muestra cómo volver a publicar un mensaje MQTT recibido de los sensores meteorológicos como un mensaje que contiene solo el ID del sensor y el valor de temperatura. Solo usaAWS IoT Corey muestra una consulta SQL simple y cómo utilizar el cliente MQTT de para probar la regla.

- [Tutorial: Envío de una notificación de Amazon SNS \(p. 201\)](#)

En este tutorial se muestra cómo enviar un mensaje SNS cuando un valor de un dispositivo sensor meteorológico supera un valor específico. Se basa en los conceptos presentados en el tutorial anterior y añade cómo trabajar con otroAWSservicio, la[Amazon Simple Notification Service](#)(Amazon SNS).

Si es la primera vez que utiliza Amazon SNS, consulte su[Introducción](#)ejercicios antes de comenzar este tutorial.

- [Tutorial: Almacenamiento de datos de dispositivos en una tabla de DynamoDB \(p. 209\)](#)

En este tutorial se muestra cómo almacenar los datos de los dispositivos sensores meteorológicos en una tabla de base de datos. Utiliza la instrucción de consulta de reglas y las plantillas de sustitución para dar formato a los datos del mensaje del servicio de destino,[Amazon DynamoDB](#).

Si es la primera vez que utiliza DynamoDB, consulte su[Introducción](#)ejercicios antes de comenzar este tutorial.

- [Tutorial: Dar formato a una notificación mediante unAWS Lambdafunción \(p. 214\)](#)

En este tutorial se muestra cómo llamar a una función Lambda para volver a formatear los datos del dispositivo y luego enviarlos como mensaje de texto. Añade un script de Python yAWSFunciones del SDK en un[AWS Lambda](#)para formatear con los datos de carga útil del mensaje de los dispositivos del sensor meteorológico y enviar un mensaje de texto.

Si es la primera vez que utiliza Lambda, consulte su[Introducción](#)ejercicios antes de comenzar este tutorial.

AWS IoTInformación general de reglas

Todos estos tutoriales creanAWS IoTreglas.

Para unAWS IoTregla para enviar los datos de un dispositivo a otroAWSservicio, utiliza:

- Una instrucción de consulta de reglas que consta de:
 - Cláusula SELECT de SQL que selecciona y da formato a los datos de la carga útil del mensaje
 - Filtro de temas (el objeto FROM de la instrucción de consulta de reglas) que identifica los mensajes que se van a utilizar
 - Una sentencia condicional opcional (una cláusula WHERE de SQL) que especifica condiciones específicas sobre las que actuar
- Tener al menos una acción de regla

Los dispositivos publican mensajes en los temas de MQTT de. El filtro de temas de la instrucción SELECT de SQL identifica los temas MQTT a los que aplicar la regla. Los campos especificados en la instrucción SELECT SQL dan formato a los datos de la carga útil del mensaje MQTT entrante para utilizarlos por las

acciones de la regla. Para obtener una lista completa de las acciones de las reglas, consulte [Acciones de las reglas de AWS IoT \(p. 479\)](#).

Tutoriales en esta sección

- [Tutorial: Republicación de un mensaje MQTT \(p. 195\)](#)
- [Tutorial: Envío de una notificación de Amazon SNS \(p. 201\)](#)
- [Tutorial: Almacenamiento de datos de dispositivos en una tabla de DynamoDB \(p. 209\)](#)
- [Tutorial: Dar formato a una notificación mediante unAWS Lambdafunción \(p. 214\)](#)

Tutorial: Republicación de un mensaje MQTT

En este tutorial se muestra cómo crear unAWS IoTque publica un mensaje MQTT cuando se recibe un mensaje MQTT especificado. La regla puede modificar la carga útil del mensaje entrante antes de publicarla. Esto permite crear mensajes adaptados a aplicaciones específicas sin necesidad de modificar su dispositivo o su firmware. También puede utilizar el aspecto de filtrado de una regla para publicar mensajes solo cuando se cumple una condición específica.

Los mensajes republicados por una regla actúan como mensajes enviados por cualquier otraAWS IoTdispositivo o cliente. Los dispositivos pueden suscribirse a los mensajes republicados de la misma manera que pueden suscribirse a cualquier otro tema de mensaje MQTT.

Lo que aprenderás en este tutorial:

- Cómo utilizar consultas y funciones SQL simples en una instrucción de consulta de reglas
- Cómo utilizar el cliente MQTT de para probar unAWS IoTRegla de

Para completar este tutorial se necesitan aproximadamente 30 minutos.

En este tutorial, hará lo siguiente:

- [Revise los temas MQTT yAWS IoTReglas de \(p. 195\)](#)
- [Paso 1: Creación de unAWS IoTRegla para volver a publicar un mensaje MQTT \(p. 196\)](#)
- [Paso 2: Pon a prueba tu nueva regla \(p. 198\)](#)
- [Paso 3: Revisar los resultados y los próximos pasos \(p. 201\)](#)

Antes de empezar este tutorial, asegúrese de que dispone de:

- [Configurar suCuenta de AWS \(p. 18\)](#)

Necesitará suCuenta de AWSyAWS IoTconsola para completar este tutorial.

- [RevisadoVer los mensajes MQTT con el cliente MQTT de AWS IoT \(p. 65\)](#)

Asegúrese de que puede utilizar el cliente MQTT de para suscribirse y publicar en un tema. Utilizará el cliente MQTT de para probar la nueva regla en este procedimiento.

Revise los temas MQTT yAWS IoTReglas de

Antes de hablar deAWS IoT, ayuda a entender el protocolo MQTT. En las soluciones de IoT, el protocolo MQTT ofrece algunas ventajas sobre otros protocolos de comunicación de red, como HTTP, lo que lo convierte en una opción popular para su uso por dispositivos IoT. En esta sección se revisan los aspectos clave de MQTT que se aplican a este tutorial. Para obtener información acerca de cómo se compara MQTT con HTTP, consulte[Selección de un protocolo para la comunicación de su dispositivo \(p. 83\)](#).

Protocolo MQTT

El protocolo MQTT utiliza un modelo de comunicación de publicación/suscripción con su host. Para enviar datos, los dispositivos publican mensajes identificados por temas en el AWS IoT Agente de mensajes. Para recibir mensajes del agente de mensajes, los dispositivos se suscriben a los temas que recibirán enviando filtros de temas en las solicitudes de suscripción al agente de mensajes. La AWS IoT Rules engine recibe mensajes MQTT del agente de mensajes de mensajes.

Reglas de AWS IoT

AWS IoT Las reglas consisten en una instrucción de consulta de regla y una o varias acciones de regla. Cuando la AWS IoT Rules engine recibe un mensaje MQTT, estos elementos actúan sobre el mensaje de la siguiente manera.

- Instrucción de consulta de reglas

La instrucción de consulta de la regla describe los temas de MQTT que se van a utilizar, interpreta los datos de la carga útil del mensaje y da formato a los datos tal como se describe en una instrucción SQL similar a las sentencias utilizadas por las bases de datos SQL populares. El resultado de la instrucción de consulta son los datos que se envían a las acciones de la regla.

- Acción de la regla

Cada acción de regla de una regla actúa sobre los datos resultantes de la instrucción de consulta de la regla. AWS IoT es compatible con [muchas acciones de reglas \(p. 479\)](#). En este tutorial, sin embargo, te concentrarás en la [Republish \(p. 540\)](#) acción de regla, que publica el resultado de la instrucción de consulta como mensaje MQTT con un tema específico.

Paso 1: Creación de una AWS IoT regla para volver a publicar un mensaje MQTT

La AWS IoT regla que crearás en este tutorial se suscribe al `device/device_id/data` Temas MQTT `device_id` es el ID del dispositivo que envió el mensaje. Estos temas se describen mediante un [Filtro de temas \(p. 99\)](#) comodín `device/+data`, donde `+` es un carácter comodín que coincide con cualquier cadena entre los dos caracteres de barra hacia delante.

Cuando la regla recibe un mensaje de un tema coincidente, vuelve a publicar `device_id/temperature` valores como nuevo mensaje MQTT con el `device/data/temp` tema.

Por ejemplo, la carga útil de un mensaje MQTT con el `device/22/data` tema se parece a esto:

```
{  
  "temperature": 28,  
  "humidity": 80,  
  "barometer": 1013,  
  "wind": {  
    "velocity": 22,  
    "bearing": 255  
  }  
}
```

La regla toma el `temperature` valor de la carga útil del mensaje y `device_id` del tema, y los vuelve a publicar como mensaje MQTT con el `device/data/temp` tema y una carga de mensajes que tiene un aspecto similar a este:

```
{  
  "device_id": "22",  
  "temperature": 28  
}
```

Con esta regla, los dispositivos que solo necesitan el ID del dispositivo y los datos de temperatura se suscriben al `device/data/temp` tema para recibir solo esa información.

Para crear una regla que vuelva a publicar un mensaje MQTT

1. AbiertolaReglashub de laAWS IoTconsola.
2. EnReglas, eligeCreary empieza a crear tu nueva regla.
3. En la parte superior deCreación de una regla:
 - a. EnNombre, introduzca el nombre de la regla. Para este tutorial, llame un nombre**republish_temp**.

Recuerda que el nombre de una regla debe ser único dentro de tu Cuenta y Región y no puede tener espacios. Hemos utilizado un carácter de guión bajo en este nombre para separar las dos palabras del nombre de la regla.

- b. EnDescripción, describa la regla.

Una descripción significativa le ayuda a recordar qué hace esta regla y por qué la creó. La descripción puede ser el mayor tiempo que sea necesario, así que sea lo más detallada posible.

4. EnInstrucción de consulta de reglasdeCreación de una regla:
 - a. EnUso de la versión de SQL, seleccione**2016-03-23**.
 - b. En el navegadorInstrucción de consulta de reglasCuadro de edición, introduzca la instrucción:

```
SELECT topic(2) as device_id, temperature FROM 'device/+/data'
```

Esta declaración:

- Escucha mensajes MQTT con un tema que coincide con eldevice/+/dataFiltro de temas.
- Selecciona el segundo elemento de la cadena de tema y lo asigna a ladevice_id.
- Selecciona el valortemperaturedesde la carga útil del mensaje y lo asigna a latemperature.

5. EnDefinir una o varias acciones:
 - a. Para abrir la lista de acciones de regla de esta regla, elijaAcción Add.
 - b. EnSelecciona una acción, eligeReplica un mensaje en unAWS IoTTema de.
 - c. En la parte inferior de la lista de acciones, elijaConfigure actionpara abrir la página de configuración de la acción seleccionada.
6. EnConfigure action:
 - a. EnTema, introduzcadevice/data/temp. Este es el tema MQTT del mensaje que publicará esta regla.
 - b. EnCalidad del servicio, elige0 - El mensaje se entrega cero o más veces.
 - c. EnElegir o crear un rol para concederAWS IoTacceso para realizar esta acción:
 - i. Elija Create Role (Crear rol). LaCreate a new role (Crear un nuevo rol)Se abre el cuadro de diálogo.
 - ii. Escriba un nombre que describa el nuevo rol. En este tutorial, utilicerepublish_role.

Al crear un nuevo rol, se crean las políticas correctas para llevar a cabo la acción de regla y se asocian al nuevo rol. Si cambia el tema de esta acción de regla o utiliza este rol en otra acción de regla, debe actualizar la política de ese rol para autorizar el nuevo tema o acción. Para actualizar un rol existente, elijaFunción de actualizaciónen esta sección.

- iii. ElegirCreación de un rolpara crear el rol y cerrar el cuadro de diálogo.
 - d. ElegirAcción Addpara añadir la acción a la regla y volver a laCreación de una regla(Se ha creado el certificado).
-
7. LaReplica un mensaje en unAWS IoTTema deaction aparece ahora enDefinir una o varias acciones.

En el mosaico de la nueva acción, a continuaciónRepublica un mensaje en unAWS IoTTema de, puede ver el tema en el que se publicará la acción de republicación.

Esta es la única acción de regla que añadirás a esta regla.

8. EnCreación de una regla, vaya a la parte inferior y elijaCrear reglapara crear la regla y completar este paso.

Paso 2: Pon a prueba tu nueva regla

Para probar la nueva regla, utilizará el cliente MQTT para publicar y suscribirse a los mensajes MQTT utilizados por esta regla.

Abra el icono[Cliente MQTT enAWS IoTconsola](#)en una ventana nueva. Esto le permitirá editar la regla sin perder la configuración de su cliente MQTT. El cliente MQTT no conserva suscripciones ni registros de mensajes si lo dejas para ir a otra página de la consola.

Para utilizar el cliente MQTT de para probar la regla

1. En el navegador[Cliente MQTT enAWS IoTconsola](#), suscribirse a los temas de entrada, en este caso,`device/+/data`.
 - a. En el Cliente MQTT, enSuscripciones, eligeSuscripción a un tema.
 - b. EnSubscription topic, introduzca el tema del filtro de temas de entrada,`device/+/data`.
 - c. Mantén el resto de los campos con la configuración predeterminada.
 - d. Elija Subscribe to topic (Suscribirse al tema).

En el navegadorSuscripcionescolumna, enPublicación de un tema,`device/+/data`aparece.

- 2. Suscríbase al tema que publicará su regla:`device/data/temp`.
 - a. UNDERSuscripciones, eligeSuscripción a un tema nuevo, y enSubscription topic, introduzca el tema del mensaje publicado de nuevo,`device/data/temp`.
 - b. Mantén el resto de los campos con la configuración predeterminada.
 - c. Elija Subscribe to topic (Suscribirse al tema).

En el navegadorSuscripcionescolumna, endispositivo/+/datos,`device/data/temp`aparece.

- 3. Publicar un mensaje en el tema de entrada con un ID de dispositivo específico,`device/22/data`. No se pueden publicar en temas MQTT que contengan caracteres comodín.
 - a. En el Cliente MQTT, enSuscripciones, eligePublicación de un tema.
 - b. En el navegadorPublicación, introduzca el nombre del tema de entrada,`device/22/data`.
 - c. Copie los datos de ejemplo que se muestran aquí y, en el cuadro de edición debajo del nombre del tema, pegue los datos de ejemplo.

```
{  
    "temperature": 28,  
    "humidity": 80,  
    "barometer": 1013,  
    "wind": {  
        "velocity": 22,  
        "bearing": 255  
    }  
}
```

- d. Para enviar su mensaje MQTT, elijaPublicación de un tema.
- 4. Revise los mensajes que se han enviado.

- a. En el Cliente MQTT, en Suscripciones, hay un punto verde junto a los dos temas a los que te suscribiste anteriormente.

Los puntos verdes indican que se han recibido uno o más mensajes nuevos desde la última vez que los miró.

- b. UNDER Suscripciones, elige dispositivo/+/datos para comprobar que la carga útil del mensaje coincide con lo que acabas de publicar y tiene este aspecto:

```
{  
    "temperature": 28,  
    "humidity": 80,  
    "barometer": 1013,  
    "wind": {  
        "velocity": 22,  
        "bearing": 255  
    }  
}
```

- c. UNDER Suscripciones, elige dispositivo/datos/temperatura para comprobar que la carga útil de mensajes republicados tiene el siguiente aspecto:

```
{  
    "device_id": "22",  
    "temperature": 28  
}
```

Observe que la `device_id` value es una cadena citada y el `temperature` valor es numérico. Esto se debe a que el `topic()` extrajo la cadena del nombre del tema del mensaje de entrada mientras que el `temperature` value utiliza el valor numérico de la carga útil del mensaje de entrada.

Si quieres hacer el `device_id` valor un valor numérico, sustituir `topic(2)` en la instrucción de consulta de regla con:

```
cast(topic(2) AS DECIMAL)
```

Tenga en cuenta que lanzar el `topic(2)` valor de un valor numérico solo funcionará si esa parte del tema contiene solo caracteres numéricos.

5. Si ve que el mensaje correcto se publicó en el dispositivo/datos/temperatura tema, luego tu regla funcionó. Consulte qué más puede obtener información sobre la acción Regla Volver a publicar en la siguiente sección.

Si no ve que el mensaje correcto se haya publicado en el dispositivo/+/datos o dispositivo/datos/temperatura temas, consulta las sugerencias de solución de problemas.

Solución de problemas de la regla de mensaje Volver a publicar

Estas son algunas cosas que debes comprobar en caso de que no veas los resultados que esperas.

- Tienes un banner de error

Si aparece un error al publicar el mensaje de entrada, corrija primero ese error. Los siguientes pasos pueden ayudarte a corregir ese error.

- No aparece el mensaje de entrada en el cliente MQTT de

Cada vez que publicas tu mensaje de entrada en eldevice/22/data, ese mensaje debería aparecer en el cliente MQTT si se suscribió aldevice/+/datafiltro de temas tal y como se describe en el procedimiento.

Cosas que hay que revisar

- Comprueba el filtro de temas al que te suscribiste

Si se ha suscrito al tema del mensaje de entrada tal y como se describe en el procedimiento, debería ver una copia del mensaje de entrada cada vez que lo publique.

Si no ves el mensaje, comprueba el nombre del tema al que te suscribiste y compáralo con el tema en el que publicaste. Los nombres de los temas distinguen entre mayúsculas y minúsculas y el tema al que se suscribió debe ser idéntico al tema al que publicó la carga útil del mensaje.

- Compruebe la función de publicación de mensajes

En el Cliente MQTT, enSuscripciones, eligedispositivo/+/datos, revisa el tema del mensaje de publicación y, a continuación, seleccionaPublicación de un tema. Debería ver que la carga útil del mensaje del cuadro de edición debajo del tema aparece en la lista de mensajes.

- No aparece el mensaje publicado de nuevo en el cliente MQTT

Para que su regla funcione, debe tener la política correcta que le autorice a recibir y volver a publicar un mensaje y debe recibir el mensaje.

Cosas que hay que revisar

- Compruebe laRegión de AWSde su cliente MQTT y de la regla que ha creado

La consola en la que ejecuta el cliente MQTT debe estar en la mismaAWSRegión como regla que ha creado.

- Compruebe el tema del mensaje de entrada en la instrucción de consulta de regla

Para que la regla funcione, debe recibir un mensaje con el nombre del tema que coincida con el filtro de temas de la cláusula FROM de la instrucción de consulta de reglas.

Compruebe la ortografía del filtro de temas en la instrucción de consulta de reglas con la del tema en el cliente MQTT. Los nombres de tema distinguen mayúsculas y minúsculas y el tema del mensaje debe coincidir con el filtro de temas de la instrucción de consulta de reglas.

- Comprobar el contenido de la carga del mensaje de entrada

Para que la regla funcione, debe buscar el campo de datos en la carga útil del mensaje declarada en la instrucción SELECT.

Revise la ortografía deltemperatureen la instrucción de consulta de reglas con la de la carga útil del mensaje en el cliente MQTT. Los nombres de los campos distinguen entre mayúsculas ytemperaturede la instrucción de consulta de regla debe ser idéntico altemperatureen la carga útil del mensaje.

Asegúrese de que el documento JSON de la carga útil del mensaje tenga el formato correcto. Si el JSON tiene algún error, como una coma que falta, la regla no podrá leerlo.

- Compruebe el tema del mensaje publicado de nuevo en la acción de regla

El tema al que publica la acción Regla Volver a publicar el nuevo mensaje debe coincidir con el tema al que se suscribió en el cliente MQTT.

Abra la regla que creó en la consola y compruebe el tema en el que la acción de regla volverá a publicar el mensaje.

- Comprobar el rol que utiliza la regla

La acción de regla debe tener permiso para recibir el tema original y publicar el nuevo tema.

Las políticas que autorizan a la regla a recibir datos de mensajes y volver a publicarlos son específicas de los temas utilizados. Si cambia el tema utilizado para volver a publicar los datos del mensaje, debe actualizar la función de la acción de la regla para actualizar su política para que coincida con el tema actual.

Si sospecha que este es el problema, edite la acción Regla Volver a publicar y cree un nuevo rol. Los nuevos roles creados por la acción de regla reciben las autorizaciones necesarias para llevar a cabo estas acciones.

Paso 3: Revisar los resultados y los próximos pasos

En este tutorial

- Ha utilizado una consulta SQL simple y un par de funciones en una instrucción de consulta de reglas para producir un nuevo mensaje MQTT.
- Ha creado una regla que volvió a publicar ese nuevo mensaje.
- Ha utilizado el cliente MQTT de para probar elAWS IoTRegla.

Pasos siguientes

Después de volver a publicar algunos mensajes con esta regla, intente experimentar con ella para ver cómo el cambio de algunos aspectos del tutorial afecta al mensaje publicado de nuevo. A continuación mostramos algunas ideas para comenzar.

- Cambie el`device_id`en el tema del mensaje de entrada y observe el efecto en la carga útil del mensaje republicado.
- Cambie los campos seleccionados en la instrucción de consulta de reglas y observe el efecto en la carga útil del mensaje republicado.
- Prueba el siguiente tutorial de esta serie y aprende a[Tutorial: Envío de una notificación de Amazon SNS \(p. 201\)](#).

La acción Volver a publicar regla utilizada en este tutorial también puede ayudarle a depurar sentencias de consulta de reglas. Por ejemplo, puede agregar esta acción a una regla para ver cómo su instrucción de consulta de reglas da formato a los datos utilizados por sus acciones de regla.

Tutorial: Envío de una notificación de Amazon SNS

En este tutorial se muestra cómo crear unAWS IoTque envía datos de mensajes MQTT a un tema de Amazon SNS para que se puedan enviar como mensajes de texto SMS.

En este tutorial, creará una regla que envía datos de mensajes de un sensor meteorológico a todos los suscriptores de un tema de Amazon SNS, siempre que la temperatura supere el valor establecido en la regla. La regla detecta cuándo la temperatura notificada supera el valor establecido por la regla y, a continuación, crea una nueva carga útil de mensaje que incluye solo el ID del dispositivo, la temperatura notificada y el límite de temperatura que se ha superado. La regla envía la nueva carga útil de mensajes como documento JSON a un tema SNS, que notifica a todos los suscriptores del tema SNS.

Lo que aprenderás en este tutorial:

- Cómo crear y probar una notificación de Amazon SNS
- Cómo llamar a una notificación de Amazon SNS desde unAWS IoTRegla de

- Cómo utilizar consultas y funciones SQL simples en una instrucción de consulta de reglas
- Cómo utilizar el cliente MQTT de para probar unAWS IoTRegla de

Para completar este tutorial se necesitan aproximadamente 30 minutos.

En este tutorial, hará lo siguiente:

- [Paso 1: Crear un tema de Amazon SNS que envíe un mensaje de texto SMS \(p. 202\)](#)
- [Paso 2: Creación de unAWS IoTRegla para enviar el mensaje de texto \(p. 203\)](#)
- [Paso 3: Probar elAWS IoTRegla y notificación de Amazon SNS \(p. 205\)](#)
- [Paso 4: Revisar los resultados y los próximos pasos \(p. 208\)](#)

Antes de empezar este tutorial, asegúrese de que dispone de:

- [Configurar suCuenta de AWS \(p. 18\)](#)

Necesitará suCuenta de AWSyAWS IoTconsola para completar este tutorial.

- [RevisadoVer los mensajes MQTT con el cliente MQTT de AWS IoT \(p. 65\)](#)

Asegúrese de que puede utilizar el cliente MQTT de para suscribirse y publicar en un tema. Utilizará el cliente MQTT de para probar la nueva regla en este procedimiento.

- Se revisó la[Amazon Simple Notification Service](#)

Si no ha utilizado Amazon SNS con anterioridad, consulte[Configuración del acceso para Amazon SNS](#). Si ya has completado otroAWS IoTTutorial, suCuenta de AWSdebe definirse ya en una configuración correcta.

Paso 1: Crear un tema de Amazon SNS que envíe un mensaje de texto SMS

Para crear un tema de Amazon SNS que envíe un mensaje de texto SMS

1. Cree un tema de Amazon SNS.

- a. Inicie sesión en la [consola de Amazon SNS](#).
- b. En el panel de navegación izquierdo, elija Topics (Temas).
- c. En la página Temas, elija Crear tema.
- d. EnDetalles de, elige elestandar. De forma predeterminada, la consola crea un tema FIFO.
- e. EnNombre, introduzca el nombre del tema de SNS. En este tutorial, escriba **high_temp_notice**.
- f. Desplácese hasta el final de la página y elijaCrear tema.

En la consola se abre la página Detalles del nuevo tema.

2. Cree una suscripción de Amazon SNS.

Note

El número de teléfono que utilizas en esta suscripción podría incurrir en cargos por mensajería de texto de los mensajes que enviarás en este tutorial.

- a. En el navegadorhigh_temp_noticePágina de detalles del tema, elijaCrear suscripción.
- b. EnCrear suscripción, en elDetalles de, en la secciónProtocolo, elijaSMS.
- c. EnPunto de enlace, introduzca el número de teléfono que puede recibir mensajes de texto. Asegúrese de introducirlo para que empiece con un+, incluye el código de país y área y no incluye ningún otro carácter de puntuación.

- d. Elija Create subscription (Crear suscripción).
3. Pruebe la notificación de Amazon SNS.
 - a. En el navegador[Consola de Amazon SNS](#), en el panel de navegación izquierdo, elijaTemas.
 - b. Para abrir la página de detalles del tema, enTemas, en la lista de temas, elijahigh_temp_notice.
 - c. Para abrirPublicar mensaje en temaen la secciónhigh_temp_noticepágina de detalles, elijaPublicar mensaje.
 - d. EnPublicar mensaje en tema, en elCuerpo del mensajessección, enCuerpo del mensaje que se va a enviar al endpoint, escriba un mensaje breve.
 - e. Desplácese hacia abajo hasta la parte inferior de la página y elijaPublicar mensaje.
 - f. En el teléfono con el número que usaste anteriormente al crear la suscripción, confirma que se ha recibido el mensaje.

Si no has recibido el mensaje de prueba, comprueba el número de teléfono y la configuración de tu teléfono.

Asegúrese de poder publicar mensajes de prueba desde el[Consola de Amazon SNS](#)antes de continuar con el tutorial.

Paso 2: Creación de unAWS IoTregla para enviar el mensaje de texto

LaAWS IoTregla que crearás en este tutorial se suscribe aldevice/*device_id*/dataTemas MQTT*device_id*es el ID del dispositivo que envió el mensaje. Estos temas se describen en un filtro de temas como device/+data, donde la+es un carácter comodín que coincide con cualquier cadena entre los dos caracteres de barra hacia delante. Esta regla también comprueba el valor deltemperatureen la carga útil del mensaje.

Cuando la regla recibe un mensaje de un tema coincidente, toma el*device_id*del nombre del tema, eltemperaturede la carga útil del mensaje y añade un valor constante para el límite que está probando y los envía como documento JSON a un tema de notificación de Amazon SNS.

Por ejemplo, un mensaje MQTT del dispositivo sensor meteorológico número 32 utiliza eldevice/32/ datatema y tiene una carga de mensaje que tiene este aspecto:

```
{  
  "temperature": 38,  
  "humidity": 80,  
  "barometer": 1013,  
  "wind": {  
    "velocity": 22,  
    "bearing": 255  
  }  
}
```

La instrucción de consulta de reglas de la regla toma eltemperaturevalor de la carga útil del mensaje, *device_id*del nombre del tema y agrega la constantemax_temperaturevalor para enviar una carga útil de mensaje similar al tema de Amazon SNS:

```
{  
  "device_id": "32",  
  "reported_temperature": 38,  
  "max_temperature": 30  
}
```

Para crear unAWS IoTRegla para detectar un valor de temperatura de límite excesivo y crear los datos para enviarlos al tema de Amazon SNS

1. AbiertolaReglashub de laAWS IoTconsola.
2. Si es la primera regla que utiliza, elijaCrear, o bienCreación de una regla.
3. EnCreación de una regla:

- a. En Name (Nombre), escriba **temp_limit_notify**.

Recuerde que el nombre de una regla debe ser único en suCuenta de AWSy Region, y no puede tener espacios. Hemos utilizado un carácter de guión bajo en este nombre para separar las palabras del nombre de la regla.

- b. EnDescripción, describa la regla.

Una descripción significativa hace que sea más fácil recordar qué hace esta regla y por qué la creó. La descripción puede ser el mayor tiempo que sea necesario, así que sea lo más detallada posible.

4. EnInstrucción de consulta de reglasdeCreación de una regla:

- a. EnUso de la versión de SQL, seleccione2016-03-23.

- b. En el navegadorInstrucción de consulta de reglasCuadro de edición, introduzca la instrucción:

```
SELECT topic(2) as device_id,
       temperature as reported_temperature,
       30 as max_temperature
  FROM 'device/+data'
 WHERE temperature > 30
```

Esta declaración:

- Escucha mensajes MQTT con un tema que coincide con eldevice/+datafiltro de temas y que tienen untemperaturevalor superior a 30.
 - Selecciona el segundo elemento de la cadena de tema y lo asigna a ladevice_id.
 - Selecciona el valortemperaturedesde la carga útil del mensaje y lo asigna a lareported_temperature.
 - Crea un valor constante30para representar el valor límite y lo asigna a lamax_temperature.
5. Para abrir la lista de acciones de regla de esta regla, enDefinir una o varias acciones, eligeAcción Add..
 6. EnSelecciona una acción, eligeEnviar un mensaje como una notificación push SNS.
 7. Para abrir la página de configuración de la acción seleccionada, en la parte inferior de la lista de acciones, elijaConfigure action.
 8. EnConfigure action:
 - a. EnObjetivo SNS, eligeSelect, busque el tema de SNS llamadohigh_temp_notice, y elijaSelect.
 - b. EnFormato de los mensajes, eligeRAW.
 - c. EnElegir o crear un rol para concederAWS IoTacceso para realizar esta acción, eligeCreación de un rol.
 - d. EnCreate a new role (Crear un nuevo rol), enNombre, escriba un nombre exclusivo para el nuevo rol. Para este tutorial, escriba **sns_rule_role**.
 - e. Elija Create role (Crear rol).

Si vas a repetir este tutorial o reutilizar un rol existente, eligeFunción de actualizaciónnantes de continuar. Esto actualiza el documento de política del rol para que funcione con el destino de SNS.

9. ElegirAcción Add.y vuelva aCreación de una regla(Se ha creado el certificado).

En el mosaico de la nueva acción, a continuaciónEnviar un mensaje como una notificación push SNS, puede ver el tema de SNS al que llamará su regla.

Esta es la única acción de regla que añadirás a esta regla.

10. Para crear la regla y completar este paso, enCreación de una regla, vaya a la parte inferior y elijaCrear regla.

Paso 3: Probar elAWS IoTregla y notificación de Amazon SNS

Para probar la nueva regla, utilizará el cliente MQTT para publicar y suscribirse a los mensajes MQTT utilizados por esta regla.

Abra el icono[Cliente MQTT enAWS IoTconsola](#)en una ventana nueva. Esto le permitirá editar la regla sin perder la configuración de su cliente MQTT. Si deja que el cliente MQTT vaya a otra página de la consola, no conservará suscripciones ni registros de mensajes.

Para utilizar el cliente MQTT de para probar la regla

1. En el navegador[Cliente MQTT enAWS IoTconsola](#), suscribirse a los temas de entrada, en este caso,`device/+/data`.
 - a. En el Cliente MQTT, enSuscripciones, eligeSuscripción a un tema.
 - b. EnSubscription topic, introduzca el tema del filtro de temas de entrada,`device/+/data`.
 - c. Mantén el resto de los campos con la configuración predeterminada.
 - d. Elija Subscribe to topic (Suscribirse al tema).

En el navegadorSuscripcionescolumna, enPublicación de un tema,`device/+/data`aparece.

2. Publicar un mensaje en el tema de entrada con un ID de dispositivo específico,`device/32/data`. No se pueden publicar en temas MQTT que contengan caracteres comodín.
 - a. En el Cliente MQTT, enSuscripciones, eligePublicar en tema.
 - b. En el navegadorPublicación, introduzca el nombre del tema de entrada,`device/32/data`.
 - c. Copie los datos de ejemplo que se muestran aquí y, en el cuadro de edición debajo del nombre del tema, pegue los datos de ejemplo.

```
{  
    "temperature": 38,  
    "humidity": 80,  
    "barometer": 1013,  
    "wind": {  
        "velocity": 22,  
        "bearing": 255  
    }  
}
```

3. d. ElegirPublicar en temapara publicar su mensaje MQTT.
3. Confirma que se ha enviado el mensaje de texto.
 - a. En el Cliente MQTT, enSuscripciones, hay un punto verde junto al tema al que te suscribiste anteriormente.

El punto verde indica que se han recibido uno o más mensajes nuevos desde la última vez que los miró.
 - b. UNDERSuscripciones, eligedispositivo/+/datospa para comprobar que la carga útil del mensaje coincide con lo que acabas de publicar y tiene este aspecto:

```
{  
    "temperature": 38,  
    "humidity": 80,  
    "barometer": 1013,  
    "wind": {  
        "velocity": 22,  
        "bearing": 255  
    }  
}
```

- c. Comprueba el teléfono que usaste para suscribirte al tema de SNS y confirma que el contenido de la carga útil del mensaje tiene este aspecto:

```
{"device_id": "32", "reported_temperature": 38, "max_temperature": 30}
```

Observe que la `device_id` value es una cadena citada y el `temperature` valor es numérico. Esto se debe a que el `topic()` extrajo la cadena del nombre del tema del mensaje de entrada mientras que el `temperature` value utiliza el valor numérico de la carga útil del mensaje de entrada.

Si quieres hacer el `device_id` valor un valor numérico, sustituir `topic(2)` en la instrucción de consulta de regla con:

```
cast(topic(2) AS DECIMAL)
```

Tenga en cuenta que lanzar el `topic(2)` valor a un numérico, `DECIMAL` value solo funcionará si esa parte del tema contiene solo caracteres numéricos.

4. Intenta enviar un mensaje MQTT en el que la temperatura no supere el límite.
- En el Cliente MQTT, en Suscripciones, elige Publicar en tema.
 - En el navegador Publicación, introduzca el nombre del tema de entrada, `device/33/data`.
 - Copie los datos de ejemplo que se muestran aquí y, en el cuadro de edición debajo del nombre del tema, pegue los datos de ejemplo.

```
{  
    "temperature": 28,  
    "humidity": 80,  
    "barometer": 1013,  
    "wind": {  
        "velocity": 22,  
        "bearing": 255  
    }  
}
```

- d. Para enviar su mensaje MQTT, elija Publicar en tema.

Debería ver el mensaje que ha enviado en la `device/+data` suscripción. Sin embargo, dado que el valor de temperatura está por debajo de la temperatura máxima de la instrucción de consulta de reglas, no debería recibir ningún mensaje de texto.

Si no aparece el comportamiento correcto, consulte las sugerencias de solución de problemas.

Solución de problemas de la regla de mensajes SNS

Estas son algunas cosas que debes comprobar, en caso de que no veas los resultados que esperas.

- Tienes un banner de error

Si aparece un error al publicar el mensaje de entrada, corrija primero ese error. Los siguientes pasos pueden ayudarle a corregir ese error.

- No ve el mensaje de entrada en el cliente MQTT

Cada vez que publicas tu mensaje de entrada en eldevice/22/data, ese mensaje debería aparecer en el cliente MQTT, si se suscribió aldevice/+/datafiltro de temas tal y como se describe en el procedimiento.

Cosas que hay que revisar

- Comprueba el filtro de temas al que te suscribiste

Si se ha suscrito al tema del mensaje de entrada tal y como se describe en el procedimiento, debería ver una copia del mensaje de entrada cada vez que lo publique.

Si no ves el mensaje, comprueba el nombre del tema al que te suscribiste y compáralo con el tema en el que publicaste. Los nombres de los temas distinguen entre mayúsculas y minúsculas y el tema al que se suscribió debe ser idéntico al tema al que publicó la carga útil del mensaje.

- Compruebe la función de publicación de mensajes

En el Cliente MQTT, enSuscripciones, eligedispositivo/+/datos, revisa el tema del mensaje de publicación y, a continuación, seleccionaPublicar en tema. Debería ver que la carga útil del mensaje del cuadro de edición debajo del tema aparece en la lista de mensajes.

- No recibes ningún mensaje SMS

Para que su regla funcione, debe tener la política correcta que le autorice a recibir un mensaje y enviar una notificación de SNS, y debe recibir el mensaje.

Cosas que hay que revisar

- Compruebe laRegión de AWSde su cliente MQTT y de la regla que ha creado

La consola en la que ejecuta el cliente MQTT debe estar en la mismaAWSRegión como regla que ha creado.

- Compruebe que el valor de temperatura de la carga útil del mensaje supera el umbral de prueba

Si el valor de temperatura es inferior o igual a 30, tal como se define en la instrucción de consulta de reglas, la regla no realizará ninguna de sus acciones.

- Compruebe el tema del mensaje de entrada en la instrucción de consulta de regla

Para que la regla funcione, debe recibir un mensaje con el nombre del tema que coincide con el filtro de temas de la cláusula FROM de la instrucción de consulta de reglas.

Compruebe la ortografía del filtro de temas en la instrucción de consulta de reglas con la del tema en el cliente MQTT. Los nombres de tema distinguen mayúsculas y minúsculas y el tema del mensaje debe coincidir con el filtro de temas de la instrucción de consulta de reglas.

- Comprobar el contenido de la carga del mensaje de entrada

Para que la regla funcione, debe buscar el campo de datos en la carga útil del mensaje declarada en la instrucción SELECT.

Revise la ortografía deltemperatureen la instrucción de consulta de reglas con la de la carga útil del mensaje en el cliente MQTT. Los nombres de los campos distinguen entre mayúsculas ytemperaturede la instrucción de consulta de regla debe ser idéntico altemperatureen la carga útil del mensaje.

Asegúrese de que el documento JSON de la carga útil del mensaje tenga el formato correcto. Si el JSON tiene algún error, como una coma que falta, la regla no podrá leerlo.

- Compruebe el tema del mensaje publicado de nuevo en la acción de regla

El tema al que publica la acción Regla Volver a publicar el nuevo mensaje debe coincidir con el tema al que se suscribió en el cliente MQTT.

Abra la regla que creó en la consola y compruebe el tema en el que la acción de regla volverá a publicar el mensaje.

- Comprobar el rol que utiliza la regla

La acción de regla debe tener permiso para recibir el tema original y publicar el nuevo tema.

Las políticas que autorizan a la regla a recibir datos de mensajes y volver a publicarlos son específicas de los temas utilizados. Si cambia el tema utilizado para volver a publicar los datos del mensaje, debe actualizar la función de la acción de la regla para actualizar su política para que coincida con el tema actual.

Si sospecha que este es el problema, edite la acción Regla Volver a publicar y cree un nuevo rol. Los nuevos roles creados por la acción de regla reciben las autorizaciones necesarias para llevar a cabo estas acciones.

Paso 4: Revisar los resultados y los próximos pasos

En este tutorial:

- Ha creado y probado un tema de notificación de Amazon SNS y una suscripción.
- Ha utilizado una consulta SQL simple y funciones en una instrucción de consulta de reglas para crear un nuevo mensaje para la notificación.
- Ha creado unAWS IoTregla para enviar una notificación de Amazon SNS que utilizó la carga útil de mensajes personalizada.
- Ha utilizado el cliente MQTT de para probar elAWS IoTregla.

Pasos siguientes

Después de enviar algunos mensajes de texto con esta regla, prueba a experimentar con ella para ver cómo los cambios de algunos aspectos del tutorial afectan al mensaje y cuándo se envía. A continuación mostramos algunas ideas para comenzar.

- Cambie el`device_id`en el tema del mensaje de entrada y observe el efecto en el contenido del mensaje de texto.
- Cambie los campos seleccionados en la instrucción de consulta de reglas y observe el efecto en el contenido del mensaje de texto.
- Cambie la prueba de la instrucción de consulta de reglas para comprobar una temperatura mínima en lugar de una temperatura máxima. Recuerde cambiar el nombre de`max_temperature!`
- Agregue una acción de regla de republicación para enviar un mensaje MQTT cuando se envía una notificación SNS.
- Prueba el siguiente tutorial de esta serie y aprende a[Tutorial: Almacenamiento de datos de dispositivos en una tabla de DynamoDB \(p. 209\)](#).

Tutorial: Almacenamiento de datos de dispositivos en una tabla de DynamoDB

En este tutorial se muestra cómo crear unAWS IoTRegla que envía datos de mensaje a una tabla de DynamoDB.

En este tutorial, creará una regla que envía datos de mensajes de un dispositivo de sensor meteorológico imaginario a una tabla de DynamoDB. La regla da formato a los datos de muchos sensores meteorológicos de modo que se pueden agregar a una sola tabla de base de datos.

Lo que aprenderá en este tutorial

- Cómo crear una tabla de DynamoDB
- Cómo enviar datos de mensajes a una tabla de DynamoDB desde unAWS IoTRegla de
- Cómo utilizar plantillas de sustitución en unAWS IoTRegla de
- Cómo utilizar consultas y funciones SQL simples en una instrucción de consulta de reglas
- Cómo utilizar el cliente MQTT de para probar unAWS IoTRegla de

Para completar este tutorial se necesitan aproximadamente 30 minutos.

En este tutorial, hará lo siguiente:

- [Paso 1: Creación de la tabla de DynamoDB para este tutorial \(p. 209\)](#)
- [Paso 2: Creación de unAWS IoTRegla para enviar datos a la tabla de DynamoDB \(p. 210\)](#)
- [Paso 3: Probar elAWS IoTtabla de reglas y de DynamoDB \(p. 212\)](#)
- [Paso 4: Revisar los resultados y los próximos pasos \(p. 214\)](#)

Antes de empezar este tutorial, asegúrese de que dispone de:

- [Configurar suCuenta de AWS \(p. 18\)](#)

Necesitará suCuenta de AWSyAWS IoTconsola para completar este tutorial.

- [RevisadoVer los mensajes MQTT con el cliente MQTT de AWS IoT \(p. 65\)](#)

Asegúrese de que puede utilizar el cliente MQTT de para suscribirse y publicar en un tema. Utilizará el cliente MQTT de para probar la nueva regla en este procedimiento.

- Se revisó la[Amazon DynamoDB](#)resumen

Si no has usado DynamoDB antes, revisa[Introducción a DynamoDB](#)para familiarizarse con los conceptos básicos y las operaciones de DynamoDB.

Paso 1: Creación de la tabla de DynamoDB para este tutorial

En este tutorial, creará una tabla de DynamoDB con estos atributos para registrar los datos de los dispositivos sensores meteorológicos imaginarios:

- `sample_time`es una clave principal y describe la hora en que se grabó la muestra.
- `device_id`es una clave de clasificación y describe el dispositivo que proporcionó la muestra
- `device_data`son los datos recibidos del dispositivo y formateados por la instrucción de consulta de reglas

Para crear la tabla de DynamoDB para este tutorial

1. Abra el icono[Consola DynamoDB](#) luego elijaCrear tablas.
2. EnCrear tabla de DynamoDB:
 - a. EnNombre de la tabla, escriba el nombre de la tabla:**wx_data**.
 - b. EnClave principal, enClave de partición, introduzcasample_time, y en la lista de opciones situada junto al campo, elijaNumber.
 - c. ComprobarAñadir clave de ordenación.
 - d. En el campo que aparece a continuaciónAñadir clave de ordenación, introduzcadevice_id, y en la lista de opciones situada junto al campo, elijaNumber.
 - e. En la parte inferior de la página, elijaCrear.

Definirásdevice_datamás adelante, al configurar la acción de regla de DynamoDB.

Paso 2: Creación de unAWS IoTregla para enviar datos a la tabla de DynamoDB

En este paso, utilizará la instrucción de consulta de reglas para dar formato a los datos de los dispositivos de sensores meteorológicos imaginarios y escribir en la tabla de base de datos.

Una carga útil de mensaje de muestra recibida de un dispositivo sensor meteorológico tiene el siguiente aspecto:

```
{  
    "temperature": 28,  
    "humidity": 80,  
    "barometer": 1013,  
    "wind": {  
        "velocity": 22,  
        "bearing": 255  
    }  
}
```

Para la entrada de la base de datos, utilizará la instrucción de consulta de reglas para aplanar la estructura de la carga útil del mensaje y tener este aspecto:

```
{  
    "temperature": 28,  
    "humidity": 80,  
    "barometer": 1013,  
    "wind_velocity": 22,  
    "wind_bearing": 255  
}
```

En esta regla, también usarás un par de[Plantillas de sustitución \(p. 622\)](#). Las plantillas de sustitución son expresiones que permiten insertar valores dinámicos de funciones y datos de mensajes.

Para crear elAWS IoTregla para enviar datos a la tabla de DynamoDB

1. Abiertola[Reglashub de laAWS IoTconsola](#).
2. Para empezar a crear la nueva regla enReglas, eligeCrear.
3. En la parte superior deCreación de una regla:
 - a. EnNombre, introduzca el nombre de la regla,**wx_data_ddb**.

Recuerde que el nombre de una regla debe ser único en suCuenta de AWSy Region, y no puede tener espacios. Hemos utilizado un carácter de guión bajo en este nombre para separar las dos palabras del nombre de la regla.

- b. EnDescripción, describa la regla.

Una descripción significativa hace que sea más fácil recordar qué hace esta regla y por qué la creó. La descripción puede ser el mayor tiempo que sea necesario, así que sea lo más detallada posible.

4. EnInstrucción de consulta de reglasdeCreación de una regla:

- a. EnUso de la versión de SQL, seleccione**2016-03-23**.
b. En el navegadorInstrucción de consulta de reglasCuadro de edición, introduzca la instrucción:

```
SELECT temperature, humidity, barometer,  
      wind.velocity as wind_velocity,  
      wind.bearing as wind_bearing,  
   FROM 'device/+/data'
```

Esta declaración:

- Escucha mensajes MQTT con un tema que coincide con eldevice/+/dataFiltro de temas.
- Da formato a los elementos delwindatributo como atributos individuales.
- Aprueba eltemperature,humidity, ybarometeratributos sin cambios.

5. EnDefinir una o varias acciones:

- a. Para abrir la lista de acciones de regla de esta regla, elijaAcción Add..
b. EnSelecciona una acción, eligelInsertar un mensaje en una tabla de DynamoDB.
c. Para abrir la página de configuración de la acción seleccionada, en la parte inferior de la lista de acciones, elijaConfigure action.

6. EnConfigure action:

- a. EnNombre de la tabla, elija el nombre de la tabla de DynamoDB que creó en un paso anterior:**wx_data**.

LaClave de partición,Tipo de clave de partición,Sort key, yTipo de clave de ordenaciónlos campos se rellenan con los valores de la tabla de DynamoDB.

- b. EnValor de clave de partición. introduzca**#{timestamp()}**.

Esta es la primera de las[Plantillas de sustitución \(p. 622\)](#)usarás en esta regla. En lugar de utilizar un valor de la carga útil del mensaje, utilizará el valor devuelto desde [eltimestamp \(p. 613\)](#)facción.

- c. En Valor de clave de clasificación, escriba**#{cast(topic(2) AS DECIMAL)}**.

Este es el segundo de los[Plantillas de sustitución \(p. 622\)](#)usarás en esta regla. Inserta el valor del segundo elemento en[Tema de \(p. 613\)](#)name, que es el ID del dispositivo, después de [élelencos \(p. 581\)](#)a un valor DECIMAL para que coincida con el formato numérico de la clave.

- d. En Write message data to this column (Escribir datos del mensaje en esta columna) introduzca **device_data**.

Esto creará eldevice_dataen la tabla de DynamoDB.

- e. Deje Operation (Operación) en blanco.
f. EnElegir o crear un rol para concederAWS IoTacceso para realizar esta acción, eligeCreación de un rol.
g. EnCreate a new role (Crear un nuevo rol), introduzca**wx_ddb_role**, y elijaCreación de un rol.

- h. En la parte inferior deConfigure action, eligeAcción Add..
- i. Cree la regla, en la parte inferior deCreación de una regla, eligeCrear regla.

Paso 3: Probar elAWS IoTtabla de reglas y de DynamoDB

Para probar la nueva regla, utilizará el cliente MQTT para publicar y suscribirse a los mensajes MQTT utilizados en esta prueba.

Abra el icono[Cliente MQTT enAWS IoTconsola](#)en una ventana nueva. Esto le permitirá editar la regla sin perder la configuración de su cliente MQTT. El cliente MQTT no conserva suscripciones ni registros de mensajes si lo dejas para ir a otra página de la consola. También querrás abrir una ventana de consola separada en el[Centro de tablas DynamoDB en elAWS IoTconsola](#)para ver las nuevas entradas que envía la regla.

Para utilizar el cliente MQTT de para probar la regla

1. En el navegador[Cliente MQTT enAWS IoTconsola](#), suscribirse al tema de entrada,**device/+/data**.
 - a. En el cliente MQTT, elijaSuscripción a un tema.
 - b. ParaFiltro de temas, introduzca el tema del filtro de temas de entrada,**device/+/data**.
 - c. Elija Subscribe.
2. Ahora, publique un mensaje en el tema de entrada con un ID de dispositivo específico,**device/22/data**. No se pueden publicar en temas MQTT que contengan caracteres comodín.
 - a. En el cliente MQTT, elijaPublicación de un tema.
 - b. ParaNombre del tema, introduzca el nombre del tema de entrada,**device/22/data**.
 - c. ParaCarga de mensajes, introduzca los siguientes datos de ejemplo.

```
{  
    "temperature": 28,  
    "humidity": 80,  
    "barometer": 1013,  
    "wind": {  
        "velocity": 22,  
        "bearing": 255  
    }  
}
```

- d. Para publicar el mensaje MQTT, elijaPublicación.
- e. Ahora, en el cliente MQTT, elijaSuscripción a un tema. En el navegadorSuscribirsecolumna, elija la**device/+/data**suscripción. Confirme que los datos de ejemplo del paso anterior aparecen allí.
3. Compruebe la fila de la tabla de DynamoDB que creó la regla.
 - a. En el navegador[Centro de tablas DynamoDB en elAWS IoTconsola](#), eligewx_datay, a continuación, elija laElementospestaña.

Si ya estás en elElementos, puede que tenga que actualizar la pantalla seleccionando el ícono de actualización de la esquina superior derecha del encabezado de la tabla.

 - b. Observe que lasample_timelos valores de la tabla son enlaces y abre uno. Si acabas de enviar tu primer mensaje, será el único de la lista.

En este enlace se muestran todos los datos de esa fila de la tabla.

 - c. Expanda la**device_data**entrada para ver los datos resultantes de la instrucción de consulta de reglas.
 - d. Explore las distintas representaciones de los datos disponibles en esta pantalla. También puede editar los datos de esta pantalla.

- e. Una vez que haya terminado de revisar esta fila de datos, para guardar los cambios realizados, elijaGuardar, o para salir sin guardar ningún cambio, elijaCancelar.

Si no aparece el comportamiento correcto, consulte las sugerencias de solución de problemas.

Solución de problemas de la regla de DynamoDB

Estas son algunas cosas que debes comprobar en caso de que no veas los resultados que esperas.

- Tienes un banner de error

Si aparece un error al publicar el mensaje de entrada, corrija primero ese error. Los siguientes pasos pueden ayudarte a corregir ese error.

- No ve el mensaje de entrada en el cliente MQTT

Cada vez que publicas tu mensaje de entrada en eldevice/22/data, ese mensaje debería aparecer en el cliente MQTT si se suscribió aldevice/+/datafiltro de temas tal y como se describe en el procedimiento.

Cosas que hay que revisar

- Comprueba el filtro de temas al que te suscribiste

Si se ha suscrito al tema del mensaje de entrada tal y como se describe en el procedimiento, debería ver una copia del mensaje de entrada cada vez que lo publique.

Si no ves el mensaje, comprueba el nombre del tema al que te suscribiste y compáralo con el tema en el que publicaste. Los nombres de los temas distinguen entre mayúsculas y minúsculas y el tema al que se suscribió debe ser idéntico al tema al que publicó la carga útil del mensaje.

- Compruebe la función de publicación de mensajes

En el Cliente MQTT, enSuscripciones, eligedispositivo/+/datos, revisa el tema del mensaje de publicación y, a continuación, seleccionaPublicación de un tema. Debería ver que la carga útil del mensaje del cuadro de edición debajo del tema aparece en la lista de mensajes.

- No ve los datos en la tabla de DynamoDB

Lo primero que debe hacer es actualizar la pantalla seleccionando el icono de actualización de la esquina superior derecha del encabezado de la tabla. Si no muestra los datos que está buscando, comprueba lo siguiente.

Cosas que hay que revisar

- Compruebe laRegión de AWSde su cliente MQTT y de la regla que ha creado

La consola en la que ejecuta el cliente MQTT debe estar en la mismaAWSRegión como regla que ha creado.

- Compruebe el tema del mensaje de entrada en la instrucción de consulta de regla

Para que la regla funcione, debe recibir un mensaje con el nombre del tema que coincide con el filtro de temas de la cláusula FROM de la instrucción de consulta de reglas.

Compruebe la ortografía del filtro de temas en la instrucción de consulta de reglas con la del tema en el cliente MQTT. Los nombres de tema distinguen mayúsculas y minúsculas y el tema del mensaje debe coincidir con el filtro de temas de la instrucción de consulta de reglas.

- Comprobar el contenido de la carga del mensaje de entrada

Para que la regla funcione, debe buscar el campo de datos en la carga útil del mensaje declarada en la instrucción SELECT.

Revise la ortografía deltemperaturreen la instrucción de consulta de reglas con la de la carga útil del mensaje en el cliente MQTT. Los nombres de los campos distinguen entre mayúsculas ytemperaturrede la instrucción de consulta de regla debe ser idéntico altemperaturreen la carga útil del mensaje.

Asegúrese de que el documento JSON de la carga útil del mensaje tenga el formato correcto. Si el JSON tiene algún error, como una coma que falta, la regla no podrá leerlo.

- Compruebe los nombres de clave y campo utilizados en la acción de regla

Los nombres de campo utilizados en la regla de tema deben coincidir con los que se encuentran en la carga útil del mensaje JSON del mensaje publicado.

Abra la regla que creó en la consola y compruebe los nombres de los campos en la configuración de acciones de reglas con los utilizados en el cliente MQTT.

- Comprobar el rol que utiliza la regla

La acción de regla debe tener permiso para recibir el tema original y publicar el nuevo tema.

Las políticas que autorizan a la regla a recibir datos de mensajes y actualizar la tabla de DynamoDB son específicas de los temas utilizados. Si cambia el tema o el nombre de la tabla de DynamoDB utilizado por la regla, debe actualizar la función de la acción de regla para actualizar su política para que coincida.

Si sospecha que este es el problema, edite la acción de la regla y cree un nuevo rol. Los nuevos roles creados por la acción de regla reciben las autorizaciones necesarias para llevar a cabo estas acciones.

Paso 4: Revisar los resultados y los próximos pasos

Después de enviar algunos mensajes a la tabla de DynamoDB con esta regla, intente experimentar con ella para ver cómo el cambio de algunos aspectos del tutorial afecta a los datos escritos en la tabla. A continuación mostramos algunas ideas para comenzar.

- Cambie eldevice_iden el tema del mensaje de entrada y observa el efecto sobre los datos. Puede utilizarlo para simular la recepción de datos de varios sensores meteorológicos.
- Cambie los campos seleccionados en la instrucción de consulta de reglas y observe el efecto en los datos. Puede utilizar esta opción para filtrar los datos almacenados en la tabla.
- Agregue una acción de regla de republicación para enviar un mensaje MQTT para cada fila añadida a la tabla. Podrías usar esto para depurar.

Una vez completado este tutorial, consulte[the section called “Dar formato a una notificación mediante unAWS Lambdafunción” \(p. 214\)](#).

Tutorial: Dar formato a una notificación mediante unAWS Lambdafunción

En este tutorial, se muestra cómo enviar datos de mensajes MQTT de a unAWS Lambdaacción para formatear y enviar a otroAWSservicio. En este tutorial, laAWS Lambdaaction utiliza laAWSSDK para enviar el mensaje con formato al tema de Amazon SNS que creó en el tutorial sobre cómo[the section called “Envío de una notificación de Amazon SNS” \(p. 201\)](#).

En el tutorial sobre cómo[the section called “Envío de una notificación de Amazon SNS” \(p. 201\)](#), el documento JSON resultante de la instrucción de consulta de la regla se envió como cuerpo del mensaje de texto. El resultado fue un mensaje de texto que se parecía a este ejemplo:

```
{"device_id": "32", "reported_temperature": 38, "max_temperature": 30}
```

En este tutorial, utilizará unAWS Lambdaacción de regla para llamar a unAWS Lambdaque da formato a los datos de la instrucción de consulta de reglas en un formato más amigable, como este ejemplo:

```
Device 32 reports a temperature of 38, which exceeds the limit of 30.
```

LaAWS Lambdaque creará en este tutorial formatea la cadena de mensajes utilizando los datos de la instrucción de consulta de reglas y llama al[Publicación de SNS](#)función delAWSSDK para crear la notificación.

Lo que aprenderá en este tutorial

- Cómo crear y probar unAWS Lambdafunción
- Cómo utilizar laAWSSDK en unAWS Lambdafunción para publicar una notificación de Amazon SNS
- Cómo utilizar consultas y funciones SQL simples en una instrucción de consulta de reglas
- Cómo utilizar el cliente MQTT de para probar unAWS IoTRegla de

Para completar este tutorial se necesitan aproximadamente 45 minutos.

En este tutorial, hará lo siguiente:

- [Paso 1: Creación de unAWS Lambdafunción que envía un mensaje de texto \(p. 215\)](#)
- [Paso 2: Creación de unAWS IoTRegla con unAWS LambdaAcción de la regla de \(p. 218\)](#)
- [Paso 3: Probar elAWS IoTRegla yAWS LambdaAcción de la regla de \(p. 219\)](#)
- [Paso 4: Revisar los resultados y los próximos pasos \(p. 222\)](#)

Antes de empezar este tutorial, asegúrese de que dispone de:

- [Configurar suCuenta de AWS \(p. 18\)](#)

Necesitará suCuenta de AWSyAWS IoTconsola para completar este tutorial.

- [RevisadoVer los mensajes MQTT con el cliente MQTT de AWS IoT \(p. 65\)](#)

Asegúrese de que puede utilizar el cliente MQTT de para suscribirse y publicar en un tema. Utilizará el cliente MQTT de para probar la nueva regla en este procedimiento.

- Has completado los demás tutoriales de reglas de esta sección

Este tutorial requiere el tema de notificación de SNS que creó en el tutorial sobre cómo[the section called "Envío de una notificación de Amazon SNS" \(p. 201\)](#). También supone que has completado los demás tutoriales relacionados con las reglas de esta sección.

- Se revisó laAWS Lambdaresumen

Si no ha utilizadoAWS Lambdaantes, revisa[AWS Lambda](#)y[Introducción al Lambda](#)para conocer sus términos y conceptos.

Paso 1: Creación de unAWS Lambdafunción que envía un mensaje de texto

LaAWS Lambdaen este tutorial recibe el resultado de la instrucción de consulta de reglas, inserta los elementos en una cadena de texto y envía la cadena resultante a Amazon SNS como mensaje de una notificación.

A diferencia del tutorial sobre cómo [the section called “Envío de una notificación de Amazon SNS” \(p. 201\)](#), que utilizaba un AWS IoT acción de regla para enviar la notificación, este tutorial envía la notificación desde la función Lambda mediante una función del AWSSDK. Sin embargo, el tema de notificación de Amazon SNS que se utiliza en este tutorial es el mismo que utilizó en el tutorial sobre cómo [the section called “Envío de una notificación de Amazon SNS” \(p. 201\)](#).

Para crear un AWS Lambda función que envía un mensaje de texto

1. Creación de un nuevo AWS Lambda función.
 - a. En la [consola de AWS Lambda](#), elija Create function (Crear función).
 - b. En Creación de una función, seleccione Utilice un proyecto.

Busque y selecciónelo **hello-world-python** blueprint y, a continuación, elija Configurar.

 - c. En Información básica:
 - i. En Nombre de la función, introduzca el nombre de esta función, **format-high-temp-notification**.
 - ii. En Rol de ejecución, elige Cree un nuevo rol desde AWS Plantillas de políticas.
 - iii. Escriba el nombre del nuevo rol en el campo Role name (Nombre de rol), **format-high-temp-notification-role**.
 - iv. En Plantillas de políticas -opcional, busca y selecciona Política de publicación de Amazon SNS.
 - v. Elija Create function (Crear función).
2. Modifique el código del blueprint para dar formato y enviar una notificación de Amazon SNS.
 - a. Despues de crear la función, debería ver la notificación de formato a alta temperatura Detalles de la página de. Si no, ábralo desde el [Lambda:Funciones](#) (Se ha creado el certificado).
 - b. En el navegador notificación de formato a alta temperatura en la página de detalles, elija la Configuración y desplácese hasta la Código de función panel.
 - c. En el navegador Código de función ventana, en la Entorno, elija el archivo Python, `lambda_function.py`.
 - d. En el navegador Código de función, elimine todo el código del programa original del blueprint y sustitúyalo 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. Elija Implementar.
3. En una nueva ventana, busque el Nombre de recurso de Amazon (ARN) del tema de Amazon SNS del tutorial acerca de cómo [the section called “Envío de una notificación de Amazon SNS” \(p. 201\)](#).
 - a. En una ventana nueva, abra la [Página de temas de la consola de Amazon SNS](#).
 - b. En el navegador Temas, busque la `high_temp_notice` tema de notificación de la lista de temas de Amazon SNS.
 - c. Busque el ARN del `high_temp_notice` tema de notificación para utilizar en el paso siguiente.
4. Cree un caso de prueba para la función Lambda.
 - a. En el navegador [Lambda:Funciones](#) Página de la consola, en el notificación de formato a alta temperatura página de detalles, elija Seleccione un evento de prueba en la esquina superior derecha de la página (aunque parezca desactivada) y elija Configurar eventos de prueba.
 - b. En Configurar evento de prueba, elige Crear un nuevo evento de prueba.
 - c. En Nombre del evento:, introduzca **SampleRuleOutput**.
 - d. En el editor JSON a continuación Nombre del evento:, pegue este documento JSON de ejemplo. Este es un ejemplo de lo que su AWS IoT se enviará a la función 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 la ventana que tiene el ARN del `high_temp_notice` tema de notificación y copia el valor de ARN.
- f. Reemplace el `notify_topic_arn` valor en el editor JSON con el ARN del tema de notificación. Mantenga abierta esta ventana para poder volver a utilizar este valor ARN cuando cree la AWS IoTRegla.
- g. Elija Create (Crear).
5. Pruebe la función con datos de muestra.
 - a. En el navegador notificación de formato a alta temperatura página de detalles, en la esquina superior derecha de la página, confirme que Salida de regla de muestra aparece junto al Pruebas botón. Si no lo hace, selecciónelo de la lista de eventos de prueba disponibles.
 - b. Para enviar el mensaje de salida de la regla de ejemplo a la función, elija Pruebas.

Si la función y la notificación funcionaron, recibirá un mensaje de texto en el teléfono que se suscribió a la notificación.

Si no has recibido un mensaje de texto en el teléfono, comprueba el resultado de la operación. En el navegadorCódigo de funciónpanel, en elResultado de ejecución, revise la respuesta para buscar cualquier error que se haya producido. No siga con el paso siguiente hasta que la función pueda enviar la notificación a su teléfono.

Paso 2: Creación de unAWS IoTRegla con unAWS LambdaAcción de la regla de

En este paso, utilizará la instrucción de consulta de reglas para dar formato a los datos del dispositivo sensor meteorológico imaginario para enviarlos a una función Lambda, que formateará y enviará un mensaje de texto.

Un ejemplo de la carga útil de mensajes recibida de los dispositivos meteorológicos tiene este aspecto:

```
{  
  "temperature": 28,  
  "humidity": 80,  
  "barometer": 1013,  
  "wind": {  
    "velocity": 22,  
    "bearing": 255  
  }  
}
```

En esta regla, utilizará la instrucción de consulta de reglas para crear una carga útil de mensajes para la función Lambda que tiene el siguiente aspecto:

```
{  
  "device_id": "32",  
  "reported_temperature": 38,  
  "max_temperature": 30,  
  "notify_topic_arn": "arn:aws:sns:us-east-1:57EXAMPLE833:high_temp_notice"  
}
```

Contiene toda la información que la función Lambda necesita para dar formato y enviar el mensaje de texto correcto.

Para crear elAWS IoTRegla para llamar a una función Lambda

1. Abra el icono[Reglashub de laAWS IoTconsola](#).
2. Para empezar a crear la nueva regla enReglas, eligeCrear.
3. En la parte superior deCreación de una regla:
 - a. EnNombre, introduzca el nombre de la regla,**wx_friendly_text**.
Recuerde que el nombre de una regla debe ser único en suCuenta de AWSy Region, y no puede tener espacios. Hemos utilizado un carácter de guión bajo en este nombre para separar las dos palabras del nombre de la regla.
 - b. EnDescripción, describa la regla.
Una descripción significativa hace que sea más fácil recordar qué hace esta regla y por qué la creó. La descripción puede ser el mayor tiempo que sea necesario, así que sea lo más detallada posible.
4. EnInstrucción de consulta de reglasdeCreación de una regla:
 - a. EnUso de la versión de SQL, seleccione**2016-03-23**.
 - b. En el navegadorInstrucción de consulta de reglasCuadro de edición, introduzca la instrucción:

```
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 declaración:

- Escucha mensajes MQTT con un tema que coincide con el dispositivo/+ /data filtro de temas y que tienen un valor superior a 30.
 - Selecciona el segundo elemento de la cadena de tema, lo convierte en un número decimal y, a continuación, lo asigna al dispositivo_id.
 - Selecciona el valor de temperature desde la carga útil del mensaje y lo asigna a la reported_temperature.
 - Crea un valor constante, 30, para representar el valor límite y lo asigna a la max_temperature.
 - Crea un valor constante para el notify_topic_arn.
- c. Consulte la ventana que tiene el ARN del tema de notificación y copia el valor de ARN.
 - d. Reemplazar el valor ARN (*arn:aws:sns:us-east-1:57 Ejemplo 833: High_TEMP_Notify*) en el editor de instrucciones de consulta de reglas con el ARN del tema de notificación.
5. En Definir una o varias acciones:
 - a. Para abrir la lista de acciones de regla de esta regla, elija Acción Add..
 - b. En Selecciona una acción, elige Enviar un mensaje a una función Lambda.
 - c. Para abrir la página de configuración de la acción seleccionada, en la parte inferior de la lista de acciones, elija Configure action.
 6. En Configure action:
 - a. En Nombre de la función, elige Select.
 - b. Elección de notificación de formato a alta temperatura.
 - c. En la parte inferior de Configure action, elige Acción Add..
 - d. Cree la regla, en la parte inferior de Creación de una regla, elige Crear regla.

Paso 3: Probar el AWS IoTregla y AWS LambdaAcción de la regla de

Para probar la nueva regla, utilizará el cliente MQTT para publicar y suscribirse a los mensajes MQTT utilizados por esta regla.

Abra el icono [Cliente MQTT en AWS IoTconsola](#) en una ventana nueva. Ahora puede editar la regla sin perder la configuración de su cliente MQTT. Si dejas el cliente MQTT para ir a otra página de la consola, perderás tus suscripciones o registros de mensajes.

Para utilizar el cliente MQTT de para probar la regla

1. En el navegador [Cliente MQTT en AWS IoTconsola](#), suscribirse a los temas de entrada, en este caso, dispositivo/+ /data.
 - a. En el Cliente MQTT, en Suscripciones, elige Suscripción a un tema.
 - b. En Subscription topic, introduzca el tema del filtro de temas de entrada, dispositivo/+ /data.
 - c. Mantén el resto de los campos con la configuración predeterminada.
 - d. Elija Subscribe to topic (Suscribirse al tema).

En el navegadorSuscripcionescolumna, enPublicación de un tema,**device/+/data** aparece.

2. Publicar un mensaje en el tema de entrada con un ID de dispositivo específico,**device/32/data**. No se pueden publicar en temas MQTT que contengan caracteres comodín.
 - a. En el Cliente MQTT, enSuscripciones, eligePublicación de un tema.
 - b. En el navegadorPublicación, introduzca el nombre del tema de entrada,**device/32/data**.
 - c. Copie los datos de ejemplo que se muestran aquí y, en el cuadro de edición debajo del nombre del tema, pegue los datos de ejemplo.

```
{  
    "temperature": 38,  
    "humidity": 80,  
    "barometer": 1013,  
    "wind": {  
        "velocity": 22,  
        "bearing": 255  
    }  
}
```

- d. Para publicar su mensaje MQTT, elijaPublicación de un tema.
3. Confirma que se ha enviado el mensaje de texto.
 - a. En el Cliente MQTT, enSuscripciones, hay un punto verde junto al tema al que te suscribiste anteriormente.

El punto verde indica que se han recibido uno o más mensajes nuevos desde la última vez que los miró.

- b. UNDERSuscripciones, eligedispositivo/+/datospara comprobar que la carga útil del mensaje coincide con lo que acabas de publicar y tiene este aspecto:

```
{  
    "temperature": 38,  
    "humidity": 80,  
    "barometer": 1013,  
    "wind": {  
        "velocity": 22,  
        "bearing": 255  
    }  
}
```

- c. Comprueba el teléfono que usaste para suscribirte al tema de SNS y confirma que el contenido de la carga útil del mensaje tiene este aspecto:

```
Device 32 reports a temperature of 38, which exceeds the limit of 30.
```

Si cambia el elemento ID de tema en el tema del mensaje, recuerde que se transmite eltopic(2)valor de un valor numérico solo funcionará si ese elemento del tema del mensaje contiene solo caracteres numéricos.

4. Intenta enviar un mensaje MQTT en el que la temperatura no supere el límite.
 - a. En el Cliente MQTT, enSuscripciones, eligePublicación de un tema.
 - b. En el navegadorPublicación, introduzca el nombre del tema de entrada,**device/33/data**.
 - c. Copie los datos de ejemplo que se muestran aquí y, en el cuadro de edición debajo del nombre del tema, pegue los datos de ejemplo.

```
{
```

```
"temperature": 28,  
"humidity": 80,  
"barometer": 1013,  
"wind": {  
    "velocity": 22,  
    "bearing": 255  
}  
}
```

- d. Para enviar su mensaje MQTT, elijaPublicación de un tema.

Debería ver el mensaje que ha enviado en la[device/+/data](#) suscripción; sin embargo, dado que el valor de temperatura está por debajo de la temperatura máxima de la instrucción de consulta de reglas, no debería recibir ningún mensaje de texto.

Si no aparece el comportamiento correcto, consulte las sugerencias de solución de problemas.

Solución de problemas deAWS Lambda regla y notificación

Estas son algunas cosas que debes comprobar, en caso de que no veas los resultados que esperas.

- Tienes un banner de error

Si aparece un error al publicar el mensaje de entrada, corrija primero ese error. Los siguientes pasos pueden ayudarte a corregir ese error.

- No aparece el mensaje de entrada en el cliente MQTT de

Cada vez que publicas tu mensaje de entrada en el[device/32/data](#), ese mensaje debería aparecer en el cliente MQTT, si se suscribió al[device/+/data](#) filtro de temas tal y como se describe en el procedimiento.

Cosas que hay que revisar

- Comprueba el filtro de temas al que te suscribiste

Si se ha suscrito al tema del mensaje de entrada tal y como se describe en el procedimiento, debería ver una copia del mensaje de entrada cada vez que lo publique.

Si no ves el mensaje, comprueba el nombre del tema al que te suscribiste y compáralo con el tema en el que publicaste. Los nombres de los temas distinguen entre mayúsculas y minúsculas y el tema al que se suscribió debe ser idéntico al tema al que publicó la carga útil del mensaje.

- Compruebe la función de publicación de mensajes

En el Cliente MQTT, en[Suscripciones](#), elige[dispositivo/+/datos](#), revisa el tema del mensaje de publicación y, a continuación, seleccionaPublicación de un tema. Debería ver que la carga útil del mensaje del cuadro de edición debajo del tema aparece en la lista de mensajes.

- No recibes ningún mensaje SMS

Para que su regla funcione, debe tener la política correcta que le autorice a recibir un mensaje y enviar una notificación de SNS, y debe recibir el mensaje.

Cosas que hay que revisar

- Compruebe laRegión de AWSde su cliente MQTT y de la regla que ha creado

La consola en la que ejecuta el cliente MQTT debe estar en la mismaAWSRegión como regla que ha creado.

- Compruebe que el valor de temperatura de la carga útil del mensaje supera el umbral de prueba

Si el valor de temperatura es inferior o igual a 30, tal como se define en la instrucción de consulta de reglas, la regla no realizará ninguna de sus acciones.

- Compruebe el tema del mensaje de entrada en la instrucción de consulta de regla

Para que la regla funcione, debe recibir un mensaje con el nombre del tema que coincida con el filtro de temas de la cláusula FROM de la instrucción de consulta de reglas.

Compruebe la ortografía del filtro de temas en la instrucción de consulta de reglas con la del tema en el cliente MQTT. Los nombres de tema distinguen mayúsculas y minúsculas y el tema del mensaje debe coincidir con el filtro de temas de la instrucción de consulta de reglas.

- Comprobar el contenido de la carga del mensaje de entrada

Para que la regla funcione, debe buscar el campo de datos en la carga útil del mensaje declarada en la instrucción SELECT.

Revise la ortografía del `temperature` en la instrucción de consulta de reglas con la de la carga útil del mensaje en el cliente MQTT. Los nombres de los campos distinguen entre mayúsculas y `temperature` de la instrucción de consulta de regla debe ser idéntico al `temperature` en la carga útil del mensaje.

Asegúrese de que el documento JSON de la carga útil del mensaje tenga el formato correcto. Si el JSON tiene algún error, como una coma que falta, la regla no podrá leerlo.

- Compruebe la notificación de Amazon SNS

En [Paso 1: Crear un tema de Amazon SNS que envíe un mensaje de texto SMS \(p. 202\)](#), consulte el paso 3 que describe cómo probar la notificación de Amazon SNS y probar la notificación para asegurarse de que la notificación funciona.

- Compruebe la función Lambda

En [Paso 1: Creación de un AWS Lambda función que envía un mensaje de texto \(p. 215\)](#), consulte el paso 5 que describe cómo probar la función Lambda utilizando datos de prueba y probar la función Lambda.

- Comprobar el rol que utiliza la regla

La acción de regla debe tener permiso para recibir el tema original y publicar el nuevo tema.

Las políticas que autorizan a la regla a recibir datos de mensajes y volver a publicarlos son específicas de los temas utilizados. Si cambia el tema utilizado para volver a publicar los datos del mensaje, debe actualizar la función de la acción de la regla para actualizar su política para que coincida con el tema actual.

Si sospecha que este es el problema, edite la acción Regla Volver a publicar y cree un nuevo rol. Los nuevos roles creados por la acción de regla reciben las autorizaciones necesarias para llevar a cabo estas acciones.

Paso 4: Revisar los resultados y los próximos pasos

En este tutorial:

- Has creado un AWS IoTregla para llamar a una función Lambda que envió una notificación de Amazon SNS que utilizaba la carga útil de mensajes personalizados.
- Ha utilizado una consulta SQL simple y funciones en una instrucción de consulta de reglas para crear una nueva carga útil de mensajes para la función Lambda.
- Ha utilizado el cliente MQTT de para probar su AWS IoTregla.

Pasos siguientes

Después de enviar algunos mensajes de texto con esta regla, prueba a experimentar con ella para ver cómo los cambios de algunos aspectos del tutorial afectan al mensaje y cuándo se envía. A continuación mostramos algunas ideas para comenzar.

- Cambie el `device_id` en el tema del mensaje de entrada y observe el efecto en el contenido del mensaje de texto.
- Cambie los campos seleccionados en la instrucción de consulta de reglas, actualice la función Lambda para utilizarlos en un nuevo mensaje y observe el efecto en el contenido del mensaje de texto.
- Cambie la prueba de la instrucción de consulta de reglas para comprobar una temperatura mínima en lugar de una temperatura máxima. Actualice la función Lambda para dar formato a un nuevo mensaje y recuerde cambiar el nombre de `max_temperature`.
- Para obtener más información sobre cómo encontrar errores que podrían producirse durante el desarrollo y el uso AWS IoT, consulte [Monitorización de AWS IoT \(p. 426\)](#).

Conservación del estado del dispositivo mientras el dispositivo está desconectado con Device Shadows

En estos tutoriales se muestra cómo utilizar el AWS IoT Servicio Device Shadow para almacenar y actualizar la información de estado de un dispositivo. El documento Shadow, que es un documento JSON, muestra el cambio en el estado del dispositivo en función de los mensajes publicados por un dispositivo, una aplicación local o un servicio. En este tutorial, el documento Shadow muestra el cambio en el color de una bombilla. Estos tutoriales también muestran cómo la sombra almacena esta información incluso cuando el dispositivo está desconectado de Internet, y transfiere la información de estado más reciente al dispositivo cuando vuelve a conectarse y solicita esta información.

Te recomendamos que pruebes estos tutoriales en el orden en que se muestran aquí, empezando por el AWS IoT los recursos que necesita crear y la configuración de hardware necesaria, lo que también te ayuda a aprender los conceptos de forma gradual. Estos tutoriales muestran cómo configurar y conectar un dispositivo Raspberry Pi para usarlo con AWS IoT. Si no tienes el hardware necesario, puedes seguir estos tutoriales adaptándolos a un dispositivo de tu elección o mediante [creación de un dispositivo virtual con Amazon EC2 \(p. 42\)](#).

Información general del escenario del tutorial

El escenario de estos tutoriales es una aplicación o servicio local que cambia el color de una bombilla y publica sus datos en temas de sombra reservados. Estos tutoriales son similares a la funcionalidad Device Shadow descrita en el [tutorial interactivo de introducción \(p. 20\)](#) y se implementan en un dispositivo Raspberry Pi. Los tutoriales de esta sección se centran en una única sombra clásica y muestran cómo puede acomodar sombras con nombre o varios dispositivos.

Los siguientes tutoriales le ayudarán a aprender a utilizar el AWS IoT Servicio Device Shadow.

- [Tutorial: Preparación de Raspberry Pi para ejecutar la aplicación de sombra \(p. 225\)](#)

Este tutorial muestra cómo configurar un dispositivo Raspberry Pi para conectarse con AWS IoT. También creará un AWS IoT documento de política y un recurso de cosa, descargue los certificados y, a continuación, adjunte la política a ese recurso de cosa. Para completar este tutorial se necesitan aproximadamente 30 minutos.

- [Tutorial: Instalación del SDK de dispositivos y ejecución de la aplicación de ejemplo para Device Shadows \(p. 230\)](#)

En este tutorial se muestra cómo instalar las herramientas, el software y el software necesarios AWS IoT SDK de dispositivo para Python y, a continuación, ejecute la aplicación sombra de ejemplo. Este

Este tutorial se basa en los conceptos presentados en [Connect un Raspberry Pi u otro dispositivo \(p. 56\)](#) y tarda 20 minutos en completarse.

- [Tutorial: Interacción con Device Shadow mediante la aplicación de ejemplo y el cliente de prueba MQTT \(p. 236\)](#)

En este tutorial se muestra cómo se utiliza el `shadow.py` aplicación de ejemplo y AWS IoT para observar la interacción entre AWS IoT Sombras del dispositivo y los cambios de estado de la bombilla. El tutorial también muestra cómo enviar mensajes MQTT a los temas reservados de Device Shadow. Este tutorial puede tardar 45 minutos en completarse.

AWS IoT Información general de sombra del dispositivo

Una Device Shadow es una representación virtual persistente de un dispositivo administrado por un [recurso de objetos \(p. 266\)](#) que crees en el AWS IoT Registro. El documento Shadow es un JSON o un JavaScript documento de notación que se utiliza para almacenar y recuperar la información del estado actual de un dispositivo. La sombra se puede utilizar para obtener y establecer el estado de un dispositivo sobre temas MQTT o API REST HTTP, sin tener en cuenta si el dispositivo está conectado o no a Internet.

Un documento Shadow contiene un `state` que describe estos aspectos del estado del dispositivo.

- `desired`: Las aplicaciones especifican los estados deseados de las propiedades del dispositivo actualizando el `desired` objeto.
- `reported`: Los dispositivos notifican su estado actual en el `reported` objeto.
- `delta`: AWS IoT informa de las diferencias entre el estado deseado y el notificado en el `delta` objeto.

Este es un ejemplo de un documento de estado Shadow (Shadow)

```
{  
  "state": {  
    "desired": {  
      "color": "green"  
    },  
    "reported": {  
      "color": "blue"  
    },  
    "delta": {  
      "color": "green"  
    }  
  }  
}
```

Para actualizar el documento Shadow de un dispositivo, puede utilizar el [Temas MQTT reservados \(p. 113\)](#), la [API REST de Device Shadow \(p. 653\)](#) que apoyan el GET, UPDATE, y DELETE operaciones con HTTP y el [AWS IoT CLI](#).

En el ejemplo anterior, digamos que desea cambiar la `desired` color a yellow. Para ello, envíe una solicitud al [UpdateThingShadow \(p. 654\)](#) API o publique un mensaje en la [Actualización \(p. 660\)](#) tema, `$aws/things/THING_NAME/shadow/update`.

```
{  
  "state": {  
    "desired": {  
      "color": "yellow"  
    }  
  }  
}
```

Las actualizaciones afectan únicamente a los campos especificados en la solicitud. Tras actualizar correctamente la sombra del dispositivo, AWS IoT publica el nuevo desirado en el tema `$aws/things/THING_NAME/shadow/delta`. El documento Shadow en este caso tiene este aspecto:

```
{  
  "state": {  
    "desired": {  
      "color": "yellow"  
    },  
    "reported": {  
      "color": "green"  
    },  
    "delta": {  
      "color": "yellow"  
    }  
  }  
}
```

El nuevo estado se informa a continuación a la AWS IoT Device Shadow mediante `UpdateTema` de `$aws/things/THING_NAME/shadow/update` con el siguiente mensaje JSON:

```
{  
  "state": {  
    "reported": {  
      "color": "yellow"  
    }  
  }  
}
```

Si desea obtener la información del estado actual, envíe una solicitud a la [GetThingShadow \(p. 654\)](#) API o publique un mensaje MQTT en el [Conseguir \(p. 658\)](#) tema `$aws/things/THING_NAME/shadow/get`.

Para obtener más información acerca del uso del servicio Device Shadow, consulte [Servicio Device Shadow de AWS IoT \(p. 627\)](#).

Para obtener más información acerca del uso de Device Shadows en dispositivos, aplicaciones y servicios, consulte [Uso de sombras en dispositivos \(p. 631\)](#) y [Uso de sombras en aplicaciones y servicios \(p. 634\)](#).

Para obtener información sobre cómo interactuar con AWS IoT sombras, consulte [Interacción con sombras \(p. 646\)](#).

Para obtener más información acerca de los temas reservados MQTT y las API REST HTTP, consulte [Temas MQTT de sombra de dispositivo \(p. 657\)](#) y [API REST de sombra de dispositivo \(p. 653\)](#).

Tutorial: Preparación de Raspberry Pi para ejecutar la aplicación de sombra

Este tutorial muestra cómo configurar y configurar un dispositivo Raspberry Pi y crear el AWS IoT recursos que necesita un dispositivo para conectar e intercambiar mensajes MQTT.

Note

Si planeas hacerlo en la sección llamada “Creación de un dispositivo virtual con Amazon EC2” (p. 42), puedes saltarse esta página y continuar en la sección llamada “Configuración del dispositivo” (p. 42). Creará estos recursos cuando cree su cosa virtual. Si desea utilizar un dispositivo diferente en lugar del Raspberry Pi, puede intentar seguir estos tutoriales adaptándolos a un dispositivo de su elección.

En este tutorial, aprenderá a:

- Configurar un dispositivo Raspberry Pi y configurarlo para usarlo con AWS IoT.

- Creación de un AWS IoT documento de política, que autoriza a su dispositivo a interactuar con AWS IoT servicios de .
- Creación de un recurso de objeto en AWS IoT los certificados de dispositivo X.509 y, a continuación, adjuntar el documento de política.

La cuestión es la representación virtual de su dispositivo en el AWS IoT registro. El certificado autentica el dispositivo para AWS IoT Core, y el documento de política autoriza a su dispositivo a interactuar con AWS IoT.

Cómo ejecutar este tutorial

Para ejecutar la shadow.py aplicación de ejemplo para Device Shadows, necesitará un dispositivo Raspberry Pi que se conecte a AWS IoT. Le recomendamos que siga este tutorial en el orden en que se presenta aquí, comenzando por configurar la Raspberry Pi y sus accesorios, y luego crear una política y adjuntar la política a un recurso de cosa que cree. A continuación, puede seguir este tutorial utilizando la interfaz gráfica de usuario (GUI) compatible con Raspberry Pi para abrir el AWS IoT en el navegador web del dispositivo, lo que también facilita la descarga de los certificados directamente en su Raspberry Pi para conectarse a AWS IoT.

Antes de comenzar este tutorial, asegúrese de que dispone de:

- Una Cuenta de AWS. Si no dispone de una, complete los pasos descritos en [Configurar su cuenta de AWS \(p. 18\)](#) antes de continuar. Necesitará su cuenta de AWS en la consola para completar este tutorial.
- La Raspberry Pi y sus accesorios necesarios. Necesitará:
 - UNA [Raspberry Pi 3 Modelo B](#) o modelo más reciente. Este tutorial podría funcionar en versiones anteriores de Raspberry Pi, pero no lo hemos probado.
 - [Sistema operativo Raspberry Pi \(32 bits\)](#) o posterior. Le recomendamos que utilice la versión más reciente del sistema operativo Raspberry Pi. Las versiones anteriores del sistema operativo podrían funcionar, pero no lo hemos probado.
 - Una conexión Ethernet o wifi.
 - Teclado, ratón, monitor, cables y fuentes de alimentación.

Para completar este tutorial se necesitan aproximadamente 30 minutos.

Paso 1: Configurar y configurar el dispositivo Raspberry Pi

En esta sección, configuraremos un dispositivo Raspberry Pi para usarlo con AWS IoT.

Important

Adaptar estas instrucciones a otros dispositivos y sistemas operativos puede ser difícil. Tendrás que entender el dispositivo lo suficientemente bien como para poder interpretar estas instrucciones y aplicarlas a tu dispositivo. Si encuentras dificultades, puedes probar una de las otras opciones de dispositivo como alternativa, como por ejemplo: [Creación de un dispositivo virtual con Amazon EC2 \(p. 42\)](#) o [Utilice su PC o Mac con Windows o Linux como AWS IoT dispositivo \(p. 50\)](#).

Tendrá que configurar su Raspberry Pi para que pueda iniciar el sistema operativo (SO), conectarse a Internet y permitirle interactuar con él en una interfaz de línea de comandos. También puede utilizar la interfaz de usuario gráfica (GUI) compatible con la Raspberry Pi para abrir el AWS IoT y ejecutar el resto de este tutorial.

Para configurar el Raspberry Pi

1. Inserte la tarjeta SD en la ranura de la tarjeta microSD de la Raspberry Pi. Algunas tarjetas SD vienen precargadas con un gestor de instalación que le solicita un menú para instalar el sistema operativo

después de arrancar la placa. También puede utilizar el generador de imágenes Raspberry Pi para instalar el sistema operativo en su tarjeta.

2. Connect un televisor o monitor HDMI al cable HDMI que se conecta al puerto HDMI del Raspberry Pi.
3. Connect el teclado y el ratón a los puertos USB del Raspberry Pi y, a continuación, enchufe el adaptador de corriente para arrancar la placa.

Después de arrancar la Raspberry Pi, si la tarjeta SD viene precargada con el administrador de instalación, aparece un menú para instalar el sistema operativo. Si tiene problemas para instalar el sistema operativo, puede seguir estos pasos. Para obtener más información acerca de la configuración de la Raspberry Pi, consulte[Configuración de la Raspberry Pi](#).

Si tiene problemas para configurar el Raspberry Pi:

- Compruebe si ha insertado la tarjeta SD antes de arrancar la placa. Si conectas la tarjeta SD después de arrancar la placa, es posible que no aparezca el menú de instalación.
- Asegúrese de que el televisor o el monitor estén encendidos y que haya seleccionado la entrada correcta.
- Asegúrese de utilizar el software compatible con Raspberry Pi.

Después de instalar y configurar el sistema operativo Raspberry Pi, abra el navegador web de Raspberry Pi y navegue hasta elAWS IoT Coreconsola para continuar con el resto de los pasos de este tutorial.

Si puede abrir elAWS IoT Coreconsola, eres Raspberry Pi está listo y puedes continuar[the section called "Aprovisionamiento del dispositivo enAWS IoT "](#) (p. 227).

Si tiene problemas o necesita ayuda adicional, consulte[Obtener ayuda para su Raspberry Pi](#).

Tutorial: Aprovisionamiento del dispositivo enAWS IoT

En esta sección se crea elAWS IoT Corerecursos que utilizará su tutorial.

Pasos para aprovisionar el dispositivo enAWS IoT

- [Paso 1: Creación de unAWS IoTpolítica para Device Shadow](#) (p. 227)
- [Paso 2: Cree un recurso de cosa y adjunte la política a la cosa](#) (p. 229)
- [Paso 3: Revisar los resultados y los próximos pasos](#) (p. 229)

Paso 1: Creación de unAWS IoTpolítica para Device Shadow

Los certificados X.509 autentican su dispositivo conAWS IoT Core. AWS IoT las directivas se adjuntan al certificado que permite que el dispositivo ejecuteAWS IoToperaciones, como suscribirse a temas reservados MQTT utilizados por el servicio Device Shadow. Su dispositivo presenta su certificado cuando se conecta y envía mensajes aAWS IoT Core.

En este procedimiento, creará una política de que permita que su dispositivo lleve a cabo elAWS IoToperaciones necesarias para ejecutar el programa de ejemplo. Le recomendamos que cree una política que conceda solo los permisos necesarios para llevar a cabo la tarea. Usted crea elAWS IoTprimero y, a continuación, adjuntarla al certificado de dispositivo que creará más adelante.

Para crear una política de AWS IoT

1. En el menú de la izquierda, elijaSeguridady, a continuación, elijaPolíticas. Si tu cuenta tiene políticas existentes, eligeCrear, de lo contrario, en elAún no tiene políticapágina, elijaCrear una política..
2. En la página Create a policy (Crear una política):

- a. Escriba un nombre para la política en elNombrefield (por ejemplo,[My_Device_Shadow_policy](#)). No utilice información personalmente identificable en sus nombres de política.
- b. En el documento de política, describe las acciones de conexión, suscripción, recepción y publicación que otorgan al dispositivo permiso para publicar y suscribirse a los temas reservados de MQTT.

Copie el siguiente ejemplo de política y péguela en el documento de política. Reemplazar thingname con el nombre de la cosa que vas a crear (por ejemplo, [My_light_bulb](#)), region con AWS IoT Región en la que utilizas los servicios, y account con el Cuenta de AWS número. Para obtener más información acerca de AWS IoT políticas, consulte [Políticas de AWS IoT Core \(p. 333\)](#).

```
{  
    "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-*"
    }
]
```

Paso 2: Cree un recurso de cosa y adjunte la política a la cosa

Dispositivos conectados a AWS IoT puede representarse por recursos de objetos en la AWS IoT Registry. UN recurso de objetos representa un dispositivo concreto o una entidad lógica, como la bombilla de este tutorial.

Para aprender a crear objeto en AWS IoT, siga los pasos descritos en [Crear objeto objeto \(p. 40\)](#). A continuación se indican algunos aspectos clave a tener en cuenta mientras sigue los pasos de ese tutorial:

1. Elegir Crear un solo objeto, y en el Nombre, escriba un nombre para el objeto que es el mismo que el thingname (por ejemplo, My_light_bulb) especificó cuando creó la política anteriormente.

No se puede cambiar el nombre de una cosa después de que se haya creado. Si le has dado un nombre distinto al thingname, cree un objeto nuevo con el nombre como thingname y borra lo antiguo.

Note

No utilice información personalmente identificable en su nombre de objeto. El nombre de la cosa puede aparecer en comunicaciones e informes sin cifrar.

2. Le recomendamos que descargue cada uno de los archivos de certificado en el. El certificado se ha creado en una página en una ubicación donde puedes encontrarlas fácilmente. Tendrá que instalar estos archivos para ejecutar la aplicación de ejemplo.

Recomendamos descargar los archivos en un certificado subdirectorio en su home en Raspberry Pi y nombre a cada uno de ellos con un nombre más simple, como se sugiere en la siguiente tabla.

Nombres de archivo de certificado

Archivos	Ruta de archivo
Certificado de entidad de certificación raíz	~/certs/Amazon-root-CA-1.pem
Certificado de dispositivo	~/certs/device.pem.crt
Clave privada	~/certs/private.pem.key

3. Despues de activar el certificado para habilitar las conexiones a AWS IoT, elige Asociar una política y asegúrese de que adjunta la política de que creó anteriormente (por ejemplo, **My_Device_Shadow_policy**) a la cosa.

Después de crear una cosa, puede ver su recurso de cosa que se muestra en la lista de cosas de la AWS IoT consola de .

Paso 3: Revisar los resultados y los próximos pasos

En este tutorial, aprendió a:

- Configura y configura el dispositivo Raspberry Pi.
- Creación de un AWS IoT documento de política que autoriza a su dispositivo a interactuar con AWS IoT servicios de .

- Cree un recurso de cosa y un certificado de dispositivo X.509 asociado y adjunte el documento de política a él.

Pasos siguientes

Ahora puede instalar el AWS IoTDevice SDK for Python, ejecute `shadow.py` aplicación de ejemplo y use Device Shadows para controlar el estado. Para obtener más información acerca de cómo ejecutar este tutorial, consulte [Tutorial: Instalación del SDK de dispositivos y ejecución de la aplicación de ejemplo para Device Shadows \(p. 230\)](#).

Tutorial: Instalación del SDK de dispositivos y ejecución de la aplicación de ejemplo para Device Shadows

En esta sección se muestra cómo puede instalar el software necesario y el AWS IoTDevice SDK for Python y ejecute el `shadow.py` aplicación de ejemplo para editar el documento Shadow y controlar el estado de la sombra.

En este tutorial, aprenderá a:

- Utilizar el software instalado y AWS IoTSDK de dispositivo para Python para ejecutar la aplicación de ejemplo.
- Obtenga información sobre cómo introducir un valor mediante la aplicación de ejemplo publica el valor deseado en el AWS IoTconsola de .
- Consulte el `shadow.py` aplicación de ejemplo y cómo utiliza el protocolo MQTT para actualizar el estado de la sombra.

Antes de ejecutar este tutorial:

Debe haber configurado su cuenta de AWS, configuró su dispositivo Raspberry Pi y creó un AWS IoTcosa y política que otorga al dispositivo permisos para publicar y suscribirse a los temas reservados de MQTT del servicio Device Shadow. Para obtener más información, consulte [Tutorial: Preparación de Raspberry Pi para ejecutar la aplicación de sombra \(p. 225\)](#).

También debes haber instalado Git, Python y el AWS IoTDevice SDK for Python. Este tutorial se basa en los conceptos presentados en el tutorial [Connect un Raspberry Pi u otro dispositivo \(p. 56\)](#). Si no ha probado ese tutorial, le recomendamos que siga los pasos descritos en dicho tutorial para instalar los archivos de certificado y el SDK del dispositivo y, a continuación, vuelva a este tutorial para ejecutar el `shadow.py` Aplicación de ejemplo.

En este tutorial, hará lo siguiente:

- [Paso 1: Ejecute la aplicación de ejemplo shadow.py \(p. 230\)](#)
- [Paso 2: Revisar la aplicación de ejemplo shadow.py Device SDK \(p. 233\)](#)
- [Paso 3: Solución de problemas con lashadow.pyAplicación de ejemplo \(p. 234\)](#)
- [Paso 4: Revisar los resultados y los próximos pasos \(p. 236\)](#)

Para completar este tutorial se necesitan aproximadamente 20 minutos.

[Paso 1: Ejecute la aplicación de ejemplo shadow.py](#)

Antes de ejecutar el `shadow.py` aplicación de ejemplo, necesitará la siguiente información además de los nombres y la ubicación de los archivos de certificado que instaló.

Valores de parámetros de aplicación

Parámetro	Dónde encontrar el valor
<code>your-iot-thing-Name</code>	Nombre del AWS IoT cosa que creó anteriormente en the section called “Paso 2: Cree un recurso de cosa y adjunte la política a la cosa” (p. 229). Para encontrar este valor, en el AWS IoT Consola , elige Manejary, a continuación, elija Objetos.
<code>your-iot-endpoint</code>	La <code>your-iot-endpoint</code> value tiene un formato de: <code>endpoint_id-ats.iot.region.amazonaws.com</code> , por ejemplo, <code>a3qj468EXAMPLE-ats.iot.us-west-2.amazonaws.com</code> . Para encontrar este valor: <ol style="list-style-type: none">En el navegador AWS IoT Consola, elige Manejary, a continuación, elija Objetos.Elige la cosa de IoT que has creado para tu dispositivo, <code>my_light_bulb</code>, que utilizó anteriormente y, a continuación, elija Interactuar. En la página de detalles de la cosa, su punto de enlace se muestra en la sección HTTPS.

Instale y ejecute la aplicación de ejemplo

- Desplácese hasta el directorio de aplicaciones de ejemplo.

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

- En la ventana de la línea de comandos, sustituya `your-iot-endpoint` y `your-iot-thing-Name` como se indica y ejecute 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
```

- Observe que la aplicación de ejemplo:

- Se conecta al AWS Servicio IoT para tu cuenta.
- Se suscribe a Delta events y Update responses.
- Le pide que introduzca el valor deseado en el terminal.
- Muestra una salida similar a la siguiente:

```
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

Si tiene problemas para ejecutar el shadow.pyAplicación de ejemplo, revise [the section called "Paso 3: Solución de problemas con la shadow.pyAplicación de ejemplo" \(p. 234\)](#). Para obtener información adicional que pueda ayudarle a corregir el problema, agregue el --verbosity para debuggar la línea de comandos para que la aplicación de ejemplo muestre mensajes detallados sobre lo que está haciendo.

Introduzca valores y observe las actualizaciones en el documento Shadow

Puede introducir valores en la terminal para especificar el desired valor, que también actualiza el reported valor. Supongamos que ingresa el color yellow en la terminal. La actualización del valor también se actualiza al color yellow. A continuación se muestran los mensajes que se muestran en el 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'.
```

Cuando publica esta solicitud de actualización, AWS IoT crea una sombra clásica predeterminada para el recurso de cosa. Puede observar la solicitud de actualización que publicó en el reported desired values en el AWS IoT mirando el documento Shadow del recurso de cosa que creó (por ejemplo, My_light_bulb). Para ver la actualización en el documento Shadow:

1. En el navegador AWS IoT consola, elija Manejar y luego en Objetos.
2. En la lista de objetos mostrada, seleccione el objeto de que ha creado y elija Sombras y, a continuación, elija Shadow clásico.

El documento Shadow debe tener un aspecto similar al siguiente. Los valores establecidos en el color yellow. Puedes ver estos valores en el Estado de sombra sección del documento.

```
{  
  "desired": {  
    "welcome": "aws-iot",  
    "color": "yellow"  
  },  
  "reported": {  
    "welcome": "aws-iot",  
    "color": "yellow"  
  }  
}
```

También ve un Metadatos que contiene la información de marca de hora y el número de versión de la solicitud.

La versión del documento de estado se puede utilizar para asegurarse de que actualiza la versión más reciente del documento Shadow de un dispositivo. Si envía otra solicitud de actualización, el número de versión aumenta 1. Cuando se suministra una versión con una solicitud de actualización, el servicio rechaza la solicitud con un código de respuesta de conflicto HTTP 409 si la versión actual del documento de estado no coincide con la versión suministrada.

```
{
```

```
"metadata": {  
    "desired": {  
        "welcome": {  
            "timestamp": 1620156892  
        },  
        "color": {  
            "timestamp": 1620156893  
        }  
    },  
    "reported": {  
        "welcome": {  
            "timestamp": 1620156892  
        },  
        "color": {  
            "timestamp": 1620156893  
        }  
    },  
    "version": 10  
}
```

Para obtener más información sobre el documento Shadow y observar los cambios en la información de estado, proceda al siguiente tutorial [Tutorial: Interacción con Device Shadow mediante la aplicación de ejemplo y el cliente de prueba MQTT \(p. 236\)](#) tal y como se describe en el [Paso 4: Revisar los resultados y los próximos pasos \(p. 236\)](#) sección de este tutorial. Si lo desea, también puede obtener información acerca de `lshadow.py` de ejemplo y cómo utiliza el protocolo MQTT en la siguiente sección.

Paso 2: Revisar la aplicación de ejemplo shadow.py Device SDK

En esta sección se revisa el `lshadow.py` Aplicación de ejemplo de la AWS IoT Device SDK v2 for Python que se utiliza en este tutorial. A continuación, revisaremos cómo se conecta a AWS IoT Core mediante el uso del protocolo MQTT y MQTT sobre WSS. La [AWS Tiempo de ejecución común \(AWS-CRT\)](#) proporciona soporte para protocolos de comunicación de bajo nivel y se incluye en el AWS IoT Device SDK v2 for Python.

Si bien este tutorial utiliza MQTT y MQTT sobre WSS, AWS IoT admite dispositivos que publican solicitudes HTTPS. Para ver un ejemplo de un programa Python que envía un mensaje HTTP desde un dispositivo, consulte el [Ejemplo de código HTTPS \(p. 96\)](#) Uso de Python `requests`.

Para obtener información sobre cómo tomar una decisión informada sobre qué protocolo utilizar para las comunicaciones de su dispositivo, consulte la [Selección de un protocolo para la comunicación de su dispositivo \(p. 83\)](#).

MQTT

La `lshadow.py` Ejemplo de llamadas `mtls_from_path` (se muestra aquí) en `elmqtt_connection_builder` para establecer una conexión con AWS IoT Core mediante el protocolo MQTT. `mtls_from_path` utiliza certificados X.509 y TLS v1.2 para autenticar el dispositivo. La AWS-LA biblioteca CRT gestiona los detalles de nivel inferior de esa conexión.

```
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  
)
```

- `endpoints` la tuyaAWS IoT punto final que ha pasado desde la línea de comandos y `client_ids` el ID que identifica este dispositivo de forma exclusiva en el Región de AWS.
- `cert_filepath`, `pri_key_filepath`, `yca_filepath` son las rutas de acceso al certificado y los archivos de clave privada del dispositivo y al archivo CA raíz.
- `client_bootstrap` es el objeto de tiempo de ejecución común que gestiona las actividades de comunicación de socket y se crea una instancia antes de la llamada `amqtt_connection_builder.mtls_from_path`.
- `on_connection_interrupted` y `on_connection_resumed` son funciones de devolución de llamada para llamar cuando se interrumpe y se reanuda la conexión del dispositivo.
- `clean_sessions` si se inicia una sesión nueva y persistente o, si hay alguna, volver a conectarse a una existente. `keep_alive_secs` el valor de mantener vivo, en segundos, para enviar el CONNTECT request. Se enviará automáticamente un ping en este intervalo. El servidor asume que la conexión se pierde si no recibe un ping después de 1,5 veces este valor.

`Lashadow.py` muestra también `llamawebsockets_with_default_aws_signingen` `amqtt_connection_builder` para establecer una conexión con AWS IoT Core utilizando el protocolo MQTT a través de WSS. MQTT sobre WSS también utiliza los mismos parámetros que MQTT y toma estos parámetros adicionales:

- `regiones` la AWS región de firma utilizada por la autenticación Signature V4, `y_credentials_providers` las AWS credenciales proporcionadas para utilizarlas para la autenticación. La región se transfiere desde la línea de comandos y `lcredentials_provider` se crea una instancia justo antes de la llamada `amqtt_connection_builder.websockets_with_default_aws_signing`.
- `websocket_proxy_options` las opciones de proxy HTTP, si se utiliza un host proxy. En el navegador `shadow.py` aplicación de muestra, este valor se crea una instancia justo antes de la llamada `amqtt_connection_builder.websockets_with_default_aws_signing`.

Suscríbete a temas y eventos de Sombra

`Lashadow.py` muestra intenta establecer una conexión y espera a que se conecte completamente. Si no está conectado, los comandos se ponen en cola. Una vez conectado, el ejemplo se suscribe a eventos delta y actualiza y recibe mensajes, y publica mensajes con un nivel de calidad de servicio (QoS) de 1 (`mqtt.QoS.AT LEAST_ONCE`).

Cuando un dispositivo se suscribe a un mensaje con QoS nivel 1, el agente de mensajes guarda los mensajes a los que está suscrito el dispositivo hasta que se puedan enviar al dispositivo. El agente de mensajes vuelve a enviar los mensajes hasta que recibe una PUBACK respuesta del dispositivo.

Para obtener más información acerca del protocolo MQTT, consulte [Revisar el protocolo MQTT \(p. 178\)](#) y [MQTT \(p. 85\)](#).

Para obtener más información sobre cómo se utilizan MQTT, MQTT sobre WSS, sesiones persistentes y niveles de QoS que se utilizan en este tutorial, consulte [Revisar la aplicación de ejemplo pubsub.py Device SDK \(p. 178\)](#).

Paso 3: Solución de problemas con `lashadow.py` Aplicación de ejemplo

Cuando ejecutas el `shadow.py` aplicación de ejemplo, debería ver algunos mensajes mostrados en el terminal y una solicitud para introducir `undesiredValue`. Si el programa emite un error, para depurar el error, puedes comenzar comprobando si ejecutó el comando correcto para su sistema.

En algunos casos, el mensaje de error podría indicar problemas de conexión y tener un aspecto similar a: `Host name was invalid for dns resolution` o `Connection was closed unexpectedly`. En tales casos, aquí hay algunas cosas que puedes comprobar:

- Comprobar la dirección del endpoint en el comando

Consulte el endpoint en el comando que ha introducido para ejecutar la aplicación de ejemplo, (por ejemplo, a3qEXAMPLEffp-ats.iot.us-west-2.amazonaws.com) y compruebe este valor en la AWS IoT consola.

Para comprobar si ha utilizado el valor correcto:

1. En el navegador AWS IoT consola, elige Manejar luego en Objetos.
2. Elija el objeto de que ha creado para su aplicación de ejemplo (por ejemplo, my_light_bulb) y luego en Interactuar.

En la página de detalles de la cosa, su punto de enlace se muestra en la sección HTTPS. También debe aparecer un mensaje que diga: This thing already appears to be connected.

- Comprobar la activación del certificado

Los certificados autentican su dispositivo con AWS IoT Core.

Para comprobar si el certificado está activo:

1. En el navegador AWS IoT consola, elige Manejar luego en Objetos.
2. Elija el objeto de que ha creado para su aplicación de ejemplo (por ejemplo, my_light_bulb) y luego en Seguridad.
3. Seleccione el certificado y, a continuación, en la página de detalles del certificado, elija Seleccionar el certificado y, a continuación, en la página de detalles del certificado, elija Actions.

Si está en la lista desplegable Activar, está disponible y solo puedes elegir Desactivar, el certificado está activo. Si no, elige Activar y vuelva a ejecutar el programa de ejemplo.

Si el programa sigue sin ejecutarse, compruebe los nombres de los archivos de certificado en el directorio certs folder.

- Compruebe la política adjunta al recurso de cosa

Mientras los certificados autentican el dispositivo, AWS IoT las directivas permiten que el dispositivo funcione AWS IoT operaciones, como suscribirse a temas reservados MQTT o publicar en ellos.

Para comprobar si se adjunta la política correcta:

1. Busque el certificado como se describió anteriormente y, a continuación, elija Políticas.
2. Elija la política que se muestra y compruebe si describe la connect, subscribe, receive, y publish acciones que otorgan al dispositivo permiso para publicar y suscribirse a los temas reservados de MQTT.

Para ver un ejemplo de política, consulte [Paso 1: Creación de una política para Device Shadow \(p. 227\)](#).

Si aparece mensajes de error que indican problemas para conectarse a AWS IoT, podría deberse a los permisos que estás utilizando para la política. En ese caso, le recomendamos que empiece con una política de acceso completo a AWS IoT recursos y, a continuación, vuelva a ejecutar el programa de ejemplo. Puede editar la política actual o elegir la política actual, elegir Desacoplar, a continuación, cree otra política que proporcione acceso completo y lo adjunte a su recurso de cosa. Posteriormente, puede restringir la política solo a las acciones y políticas que necesita para ejecutar el programa.

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

```
        ],
        "Resource": "*"
    }
}
```

- Comprobar la instalación del SDK de dispositivo

Si el programa sigue sin ejecutarse, puede volver a instalar el SDK del dispositivo para asegurarse de que la instalación del SDK está completa y correcta.

Paso 4: Revisar los resultados y los próximos pasos

En este tutorial, aprendió a:

- Instalar el software, las herramientas y el AWS IoTDevice SDK for Python.
- Comprenda cómo la aplicación de ejemplo, `shadow.py`, utiliza el protocolo MQTT para recuperar y actualizar el estado actual de la sombra.
- Ejecute la aplicación de ejemplo para Device Shadows y observe la actualización del documento Shadow en el AWS IoTconsola de . También aprendió a solucionar cualquier problema y corregir errores al ejecutar el programa.

Pasos siguientes

Ahora puede ejecutar el `shadow.py` aplicación de ejemplo y use Device Shadows para controlar el estado. Puede observar las actualizaciones del documento Shadow en el AWS IoTConsola y observa los eventos delta a los que responde la aplicación de muestra. Con el cliente de prueba MQTT, puede suscribirse a los temas de sombra reservados y observar los mensajes recibidos por los temas al ejecutar el programa de ejemplo. Para obtener más información acerca de cómo ejecutar este tutorial, consulte [Tutorial: Interacción con Device Shadow mediante la aplicación de ejemplo y el cliente de prueba MQTT \(p. 236\)](#).

Tutorial: Interacción con Device Shadow mediante la aplicación de ejemplo y el cliente de prueba MQTT

Para interactuar con el `shadow.py` aplicación de ejemplo, introduzca un valor en el terminal para `elDesiredValor`. Por ejemplo, puede especificar colores que se asemejan a los semáforos y AWS IoTresponde a la solicitud y actualiza los valores notificados.

En este tutorial, aprenderá a:

- Usar `shadow.py` aplicación de ejemplo para especificar los estados deseados y actualizar el estado actual de la sombra.
- Edite el documento Shadow para observar los eventos delta y cómo el `shadow.py` la aplicación de muestra responde a ella.
- Utilice el cliente de prueba MQTT para suscribirse a temas sombra y observar actualizaciones cuando ejecuta el programa de ejemplo.

Antes de ejecutar este tutorial, debe disponer de:

Configurar el Cuenta de AWS, configuró su dispositivo Raspberry Pi y creó un AWS IoTcosa y política. También debe haber instalado el software necesario, el SDK del dispositivo, los archivos de certificado y ejecutar el programa de ejemplo en el terminal. Para obtener más información, consulte los tutoriales anteriores [Tutorial: Preparación de Raspberry Pi para ejecutar la aplicación de sombra \(p. 225\)](#) y [Paso 1:](#)

Ejecute la aplicación de ejemplo shadow.py (p. 230). Debe completar estos tutoriales si no lo ha hecho ya.

En este tutorial, hará lo siguiente:

- Paso 1: Actualizar los valores deseados e informados mediante shadow.pyAplicación de ejemplo (p. 237)
- Paso 2: Ver mensajes desde la shadow.pyaplicación de ejemplo en el cliente de prueba MQTT (p. 238)
- Paso 3: Solución de errores con las interacciones de Device Shadow (p. 242)
- Paso 4: Revisar los resultados y los próximos pasos (p. 243)

Para completar este tutorial se necesitan aproximadamente 45 minutos.

Paso 1: Actualizar los valores deseados e informados mediante shadow.pyAplicación de ejemplo

En el tutorial anteriorPaso 1: Ejecute la aplicación de ejemplo shadow.py (p. 230), aprendió a observar un mensaje publicado en el documento Shadow en elAWS IoTconsola cuando introduce un valor deseado tal y como se describe en la secciónTutorial: Instalación del SDK de dispositivos y ejecución de la aplicación de ejemplo para Device Shadows (p. 230).

En el ejemplo anterior, establecemos el color deseado en yellow. Después de introducir cada valor, el terminal le pide que introduzca otrodesiredValor . Si vuelve a introducir el mismo valor (yellow), la aplicación lo reconoce y le pide que introduzca un nuevodesiredValor .

```
Enter desired value:  
yellow  
Local value is already 'yellow'.  
Enter desired value:
```

Ahora, digamos que ingresas el color green.AWS IoTresponde a la solicitud y actualiza elreportedValor paragreen. Así es como se produce la actualización cuando eldesiredEl estado es diferente de lareportedestado, causando un delta.

¿Cómo funciona elshadow.pyaplicación de ejemplo simula interacciones de Device Shadow:

1. Escriba undesiredValue (digamos)yellow en el terminal para publicar el estado deseado.
2. Como eldesiredEl estado es diferente de lareportedstate (digamos el color green), se produce un delta y la aplicación suscrita al delta recibe este mensaje.
3. La aplicación responde al mensaje y actualiza su estado a ladesiredValor,yellow.
4. A continuación, la aplicación publica un mensaje de actualización con el nuevo valor notificado del estado del dispositivo,yellow.

A continuación se muestran los mensajes que se muestran en el terminal que muestran cómo se publica la solicitud de actualización.

```
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'.
```

En el navegador AWS IoT, el documento Shadow refleja el valor actualizado agreeenpara los dosreportedydesiredy el número de versión se incrementa en 1. Por ejemplo, si el número de versión anterior se mostraba como 10, el número de versión actual se mostrará como 11.

Note

Eliminar una sombra no restablece el número de versión a 0. Verás que la versión de sombra se incrementa en 1 cuando publicas una solicitud de actualización o creas otra sombra con el mismo nombre.

Edite el documento Shadow para observar los eventos delta

Lashadow.pyapp de muestra también está suscrita aedeltay responde cuando se produce un cambio en eldesiredValor . Si lo prefiere, puede cambiar ladesiredvalor del colorred. Para ello, en elAWS IoTconsola, edite el documento Shadow haciendo clic enEditary, a continuación, configure eldesiredValor parareden el JSON, manteniendo elreportedValor paragreen. Antes de guardar los cambios, mantenga abierto el terminal de la Raspberry Pi, ya que verá los mensajes mostrados en el terminal cuando se produzca el cambio.

```
{  
  "desired": {  
    "welcome": "aws-iot",  
    "color": "red"  
  },  
  "reported": {  
    "welcome": "aws-iot",  
    "color": "green"  
  }  
}
```

Después de guardar el nuevo valor, elshadow.pyaplicación de muestra responde a este cambio y muestra mensajes en el terminal que indican el delta. A continuación, debería ver que aparecen los siguientes mensajes debajo de la solicitud para introducir eldesiredValor .

```
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'.
```

Paso 2: Ver mensajes desde lashadow.pyaplicación de ejemplo en el cliente de prueba MQTT

Puede utilizar elCliente de pruebas MQTTLaAWS IoTconsolapara supervisar los mensajes MQTT que se transmiten en suCuenta de AWS. Al suscribirse a temas MQTT reservados utilizados por el servicio Device Shadow, puede observar los mensajes recibidos por los temas al ejecutar la aplicación de ejemplo.

Si aún no ha utilizado el cliente de prueba MQTT, puede revisarVer los mensajes MQTT con el cliente MQTT de AWS IoT (p. 65). Esto le ayuda a aprender a utilizar elCliente de pruebas MQTTLaAWS IoTconsolapara ver mensajes MQTT a medida que pasan por el agente de mensajes.

1. Abra el cliente de prueba MQTT

Abra el iconoCliente de prueba MQTT enAWS IoTconsolaen una nueva ventana para que pueda observar los mensajes recibidos por los temas de MQTT sin perder la configuración de su cliente de prueba MQTT. El cliente de prueba MQTT no conserva suscripciones ni registros de mensajes si lo dejas para ir a otra página de la consola. Para esta sección del tutorial, puede tener el documento

Shadow de su AWS IoT cosa y el cliente de prueba MQTT se abren en ventanas separadas para observar más fácilmente la interacción con Device Shadows.

2. Suscríbase a los temas Shadow reservados de MQTT

Puede utilizar el cliente de prueba MQTT para introducir los nombres de los temas reservados de MQTT de Device Shadow y suscribirse a ellos para recibir actualizaciones al ejecutar elshadow.pyAplicación de ejemplo. Para suscribirse a los temas:

- a. En el navegadorCliente de pruebas MQTTeen laAWS IoTconsola, eligeSuscripción a un tema.
- b. En el navegadorFiltro de temassección, introduzca:\$aws/things/**thingName**/shadow/update/#. Aquí,thingnamees el nombre del recurso de objeto que creó anteriormente (por ejemplo,My_light_bulb).
- c. Mantenga los valores predeterminados para los ajustes de configuración adicionales y después elijaSuscribirse.

Mediante el uso de la#comodín en la suscripción al tema, puede suscribirse a varios temas MQTT al mismo tiempo y observar todos los mensajes que se intercambian entre el dispositivo y su sombra en una sola ventana. Para obtener más información acerca de los caracteres comodín y su uso, consulte[Temas MQTT \(p. 98\)](#).

3. Ejecución de shadow.pyprograma de ejemplo y mensajes de observación

En la ventana de línea de comandos de Raspberry Pi, si has desconectado el programa, vuelve a ejecutar la aplicación de muestra y mira los mensajes en elCliente de pruebas MQTTeen laAWS IoTconsola.

- a. Ejecute el siguiente comando para reiniciar el programa de ejemplo. Reemplazar`your-iot-thing-Name`y`your-iot-endpoint`con los nombres de laAWS IoTcosa que creó anteriormente (por ejemplo,My_light_bulb) y el punto final para interactuar con el 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
```

Lashadow.pyla aplicación de muestra se ejecuta y recupera el estado de sombra actual. Si ha eliminado la sombra o ha borrado los estados actuales, el programa establece el valor actual enoff, a continuación, le pide que ingrese undesiredValor .

```
Connecting to a3qEXAMPLEfffp-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 otro lado, si el programa se estaba ejecutando y lo reiniciaste, verás el último valor de color informado en el terminal. En el cliente de prueba MQTT, verás una actualización de los temas\$aws/things/**thingName**/sombra/gety\$aws/things/**thingName**/shadow/get/accepted.

Supongamos que el último color reportado fue green. A continuación se muestra el contenido de la \$aws/things/**thingName**/shadow/get/acceptedJSON.

```
{  
  "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. Escriba undesiredvalor en la terminal, como yellow. La aplicación de muestra responde y muestra los siguientes mensajes en el terminal que muestran el cambio en el reportedValor parayellow.

```
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'.
```

En el navegador Cliente de pruebas MQTT en la AWS IoT consola, en Suscripciones, verá que los siguientes temas han recibido un mensaje:

- \$aws/things/**thingName**/shadow/update: muestra que ambos desired y updated los valores cambian al color yellow.
- \$aws/things/**thingName**/shadow/update/accepted: muestra los valores actuales del desired y reported estados y sus metadatos e información de versión.
- \$aws/things/**thingName**/shadow/update/documents: muestra los valores anteriores y actuales del desired y reported estados y sus metadatos e información de versión.

Como el documento \$aws/things/**thingName**/shadow/update/document también contiene información contenida en los otros dos temas, podemos revisarla para ver la información del estado. El estado anterior muestra el valor notificado establecido en **green**, sus metadatos e información de versión y el estado actual que muestra el valor notificado actualizado a **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": 1617297888  
        }  
      }  
    }  
  }  
}
```

```
        "timestamp": 1617297904
    }
},
"version": 11
},
"timestamp": 1617297904
}
```

- c. Ahora, si ingresas a otrodesirevalor, verá más cambios en elreportedvalores y actualizaciones de mensajes recibidas por estos temas. El número de versión también aumenta en 1. Por ejemplo, si especifica el valorgreen, el estado anterior informa del valoryellow y el estado actual informa del valorgreen.
4. Editar documento Shadow para observar eventos delta

Para observar los cambios en el tema delta, edite el documento Shadow en elAWS IoTconsola de . Si lo prefiere, puede cambiar ladesiredvalor del colorred. Para ello, en elAWS IoTconsola, elijaEditary, a continuación, configure eldesiredvalor en rojo en el JSON, manteniendo elreportedvalor establecido engreen. Antes de guardar el cambio, mantenga el terminal abierto, ya que verá el mensaje delta informado en la terminal.

```
{
"desired": {
    "welcome": "aws-iot",
    "color": "red"
},
"reported": {
    "welcome": "aws-iot",
    "color": "green"
}
}
```

Lashadow.py aplicación de muestra responde a este cambio y muestra mensajes en el terminal que indican el delta. En el cliente de prueba MQTT,update los temas habrán recibido un mensaje en el que se muestran los cambios en eldesiredyreportedValores.

También verá que el tema\$aws/things/**thingName**/shadow/update/delta ha recibido un mensaje. Para ver el mensaje, elija este tema, que aparece enSuscripciones.

```
{
"version": 13,
"timestamp": 1617318480,
"state": {
    "color": "red"
},
"metadata": {
    "color": {
        "timestamp": 1617318480
    }
}
}
```

Paso 3: Solución de errores con las interacciones de Device Shadow

Al ejecutar la aplicación de ejemplo Shadow, es posible que surjan problemas al observar las interacciones con el servicio Device Shadow.

Si el programa se ejecuta correctamente y le pide que ingrese undesired, debería poder observar las interacciones de Device Shadow utilizando el documento Shadow y el cliente de prueba MQTT como se

describió anteriormente. Sin embargo, si no puede ver las interacciones, puede consultar algunas cosas que puede comprobar:

- Comprueba el nombre de la cosa y su sombra en elAWS IoTconsola

Si no ves los mensajes en el documento Shadow, revisa el comando y asegúrate de que coincida con el nombre de la cosa en elAWS IoTconsola. También puedes comprobar si tienes una sombra clásica eligiendo el recurso de tu cosa y luego eligiendoSombras. Este tutorial se centra principalmente en las interacciones con la sombra clásica.

También puede confirmar que el dispositivo que utilizó está conectado o no a Internet. En el navegadorAWS IoTconsola, elija el objeto de que creó anteriormente y, a continuación, elijaInteractuar. En la página de detalles de la cosa, debería ver un mensaje aquí que diga:*This thing already appears to be connected.*

- Consulta los temas reservados de MQTT a los que te suscribiste

Si no ves que los mensajes aparecen en el cliente de prueba MQTT, comprueba si los temas a los que te suscribiste tienen el formato correcto. Los temas de MQTT Device Shadow tienen un formato\$aws/things/**thingName**/shadow/y podría haberupdate,get, o biendeleteseguirlo en función de las acciones que deseé ejecutar en la sombra. En este tutorial se utiliza el tema\$aws/things/**thingName**/sombra/#así que asegúrate de haberlo introducido correctamente al suscribirte al tema en elFiltro de temassección del cliente de prueba.

Al introducir el nombre del tema, asegúrese de que el**thingName**es el mismo que el deAWS IoTcosa que creó antes. También puede suscribirse a temas MQTT adicionales para ver si se ha realizado correctamente una actualización. Por ejemplo, puedes suscribirte al tema\$aws/things/**thingName**/shadow/update/rejectedpara recibir un mensaje cada vez que falla una solicitud de actualización para poder depurar problemas de conexión. Para obtener más información acerca de los temas reservados, consulte[the section called “Temas de sombra” \(p. 113\)](#)y[Temas MQTT de sombra de dispositivo \(p. 657\)](#).

Paso 4: Revisar los resultados y los próximos pasos

En este tutorial, aprendió a:

- Usarshadow.py aplicación de ejemplo para especificar los estados deseados y actualizar el estado actual de la sombra.
- Edite el documento Shadow para observar los eventos delta y cómo elshadow.py la aplicación de muestra responde a ella.
- Utilice el cliente de prueba MQTT para suscribirse a temas sombra y observar actualizaciones cuando ejecuta el programa de ejemplo.

Pasos siguientes

Puede suscribirse a temas reservados adicionales de MQTT para observar las actualizaciones de la aplicación sombra. Por ejemplo, si solo te suscribes al tema\$aws/things/**thingName**/shadow/update/accepted, solo verá la información de estado actual cuando se haya realizado correctamente una actualización.

También puede suscribirse a temas de sombra adicionales para depurar problemas u obtener más información sobre las interacciones de Device Shadow y también depurar cualquier problema relacionado con las interacciones de Device Shadow. Para obtener más información, consulte [the section called “Temas de sombra” \(p. 113\)](#) y [Temas MQTT de sombra de dispositivo \(p. 657\)](#).

También puede optar por ampliar la aplicación utilizando sombras con nombre o utilizando hardware adicional conectado con Raspberry Pi para los LED y observar cambios en su estado mediante mensajes enviados desde el terminal.

Para obtener más información sobre el servicio Device Shadow y el uso del servicio en dispositivos, aplicaciones y servicios, consulte [Servicio Device Shadow de AWS IoT \(p. 627\)](#), [Uso de sombras en dispositivos \(p. 631\)](#), y [Uso de sombras en aplicaciones y servicios \(p. 634\)](#).

Tutorial: Creación de un autorizador personalizado paraAWS IoT Core

En este tutorial se muestran los pasos para crear, validar y utilizar la autenticación personalizada mediante el AWS CLI. Opcionalmente, con este tutorial, puede utilizar Postman para enviar datos a AWS IoT Core mediante la API de publicación HTTP.

En este tutorial se muestra cómo crear una función Lambda de ejemplo que implementa la lógica de autorización y autenticación y un autorizador personalizado mediante el `create-authorizer` llamada con firma de tokens habilitada. A continuación, el autorizador se valida mediante el `test-invoke-authorizer`, por último, puedes enviar datos a AWS IoT Core mediante la API de publicación HTTP para probar un tema de MQTT. La solicitud de ejemplo especificará el autorizador que se va a invocar mediante el `x-amz-customauthorizer-name` encabezado y pasa el nombre de la clave de tokens `x-amz-customauthorizer-signature` en los encabezados de solicitudes.

Lo que aprenderás en este tutorial:

- Cómo crear una función Lambda para que sea un gestor de autorizador personalizado
- Cómo crear un autorizador personalizado a través de la AWS CLI con la firma de token activada
- Cómo probar su autorizador personalizado mediante el `test-invoke-authorizer` comando
- Cómo publicar un tema MQTT mediante [Postman](#) y valide la solicitud con su autorizador personalizado

Este tutorial tarda aproximadamente 60 minutos en completarse.

En este tutorial, hará lo siguiente:

- [Paso 1: Elija una función de Lambda para su autorizador personalizado \(p. 245\)](#)
- [Paso 2: Cree un key pair públicas y privadas para su autorizador personalizado \(p. 248\)](#)
- [Paso 3: Crear un recurso de autorización de clientes y su autorización \(p. 248\)](#)
- [Paso 4: Probar el autorizador llamando `test-invoke-authorizer` \(p. 251\)](#)
- [Paso 5: Probar la publicación de mensajes MQTT con Postman \(p. 252\)](#)
- [Paso 6: Ver mensajes en el cliente de prueba MQTT \(p. 254\)](#)
- [Paso 7: Revisar los resultados y los próximos pasos \(p. 255\)](#)
- [Paso 8: Eliminar recursos \(p. 255\)](#)

Antes de comenzar este tutorial, asegúrese de que dispone de:

- [Configurar suCuenta de AWS \(p. 18\)](#)

Necesitará suCuenta de AWSyAWS IoTconsola para completar este tutorial.

La cuenta que utilizas para este tutorial funciona mejor cuando incluye al menos estos AWS polícticas administradas:

- [IAMFullAccess](#)
- [AWSIoTFullAccess](#)
- [AWSLambda_FullAccess](#)

Important

Las políticas de IAM utilizadas en este tutorial son más permisivas de lo que debería seguir en una implementación de producción. En un entorno de producción, asegúrese de que las políticas de cuentas y recursos otorguen solo los permisos necesarios.

Cuando se crean políticas de IAM para producción, determine qué acceso necesitan los usuarios y roles y, a continuación, diseñe las políticas que les permitan realizar solo esas tareas.

Para obtener más información, consulte [Prácticas recomendadas de seguridad en IAM](#)

- Instaló el AWS CLI

Para obtener información acerca de cómo instalar la AWS CLI, consulte [Instalación de AWS CLI](#). Este tutorial requiere AWS CLI versión aws-cli/2.1.3 Python/3.7.4 Darwin/18.7.0 exe/x86_640 posterior.

- Herramientas OpenSSL

Los ejemplos de este tutorial utilizan LibreSSL 2.6.5. También puede utilizar OpenSSL v1.1.1i herramientas para este tutorial.

- Se revisó el [AWS Lambda](#) resumen

Si no ha utilizado AWS Lambda antes, revisa [AWS Lambda](#) y [Introducción al Lambda](#) para conocer sus términos y conceptos.

- Se ha revisado cómo crear solicitudes en Postman

Para obtener más información, consulte [Solicitudes de construcción](#).

- Se han eliminado los autorizadores personalizados del tutor anterior

Su cuenta de AWS solo puede tener un número limitado de autorizadores personalizados configurados a la vez. Para obtener más información sobre cómo eliminar un autorizador personalizado, consulte [the section called "Paso 8: Eliminar recursos" \(p. 255\)](#).

Paso 1: Elija una función de Lambda para su autorizador personalizado

Autenticación personalizada en AWS IoT Core utiliza [recursos de autorizadores](#) que crea para autenticar y autorizar clientes. La función que creará en esta sección autentica y autoriza a los clientes a medida que se conectan a AWS IoT Core y acceden a AWS IoT de AWS.

La función Lambda hace lo siguiente:

- Si una solicitud viene de testInvokeAuthorizer, devuelve una política de IAM con un Deny action.
- Si una solicitud procede de Passport mediante HTTP y el actionToken tiene un valor de allow, devuelve una política de IAM con un Allow action. En caso contrario, devuelve una política de IAM con un Deny action.

Para crear la función Lambda para su autorizador personalizado

1. En el navegador [Lambda](#): consola, abierta [Funciones](#).
2. Elija Create function (Crear función).
3. Confirmar Author from scratch está seleccionado.
4. En Basic information:
 - a. Bajo Function name (Nombre de función), escriba **custom-auth-function**.

- b. En Runtime (Tiempo de ejecución);, confirmeNode.js 14.x
5. Elija Create function (Crear función).

Lambda crea una función Node.js y un [rol de ejecución](#) que otorga a la función permiso para cargar los registros. La función de Lambda asume el rol de ejecución cuando se invoca la función y utiliza el rol de ejecución para crear credenciales para el AWSSDK y para leer datos de fuentes de eventos.

6. Para ver el código y la configuración de la función en la [AWS Cloud9](#)editor, elige la función de autenticación personalizada en la ventana del diseñador y, a continuación, elija index.jsen el panel de navegación del editor.

Para lenguajes de scripting como Node.js, Lambda incluye una función básica que devuelve una respuesta de éxito. Puede utilizar el [AWS Cloud9](#)editor para editar la función siempre y cuando el código fuente no supere los 3 MB.

7. Sustituya el index.jscódigo del editor con el siguiente código:

```
// 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. Elija Implementar.
9. DespuesCambios implementadosaparece encima del editor:
 - a. Desplazar hasta ellInformación general de funcionessección sobre el editor.
 - b. Copie elARN de la función dey guárdelo para utilizarlo más adelante en este tutorial.
10. Comprobación de la función de .
 - a. Elija el iconoPruebaspestaña.
 - b. Usando la configuración de prueba predeterminada, elijalInvocar.
 - c. Si la prueba se ha realizado correctamente, en elResultados de ejecución, abraDetalles devista. Debería ver el documento de política que devolvió la función.

Si la prueba ha fallado o no ve un documento de política, revise el código para buscar y corregir los errores.

Paso 2: Cree un key pair públicas y privadas para su autorizador personalizado

El autorizador personalizado requiere una clave pública y privada para autenticarlo. Los comandos de esta sección utilizan herramientas OpenSSL para crear este key pair.

Para crear el key pair públicas y privadas para el autorizador personalizado

1. Elija el archivo de clave privada.

```
openssl genrsa -out private-key.pem 4096
```

2. Verifique el archivo de clave privada que acaba de crear.

```
openssl rsa -check -in private-key.pem -noout
```

Si el comando no muestra ningún error, el archivo de clave privada es válido.

3. Elija el archivo de clave pública.

```
openssl rsa -in private-key.pem -pubout -out public-key.pem
```

4. Verifique el archivo de clave pública.

```
openssl pkey -inform PEM -pubin -in public-key.pem -noout
```

Si el comando no muestra ningún error, el archivo de clave pública es válido.

Paso 3: Crear un recurso de autorización de clientes y su autorización

La AWS IoT autorizador personalizado es el recurso que une todos los elementos creados en los pasos anteriores. En esta sección, creará un recurso de autorización personalizado y le dará permiso para ejecutar la función Lambda que creó anteriormente. Puede crear un recurso de autorización personalizado mediante la AWS IoT consola de AWS CLI, o el AWS API.

Para este tutorial, solo tiene que crear un autorizador personalizado. En esta sección se describe cómo crear mediante la AWS IoT consola y AWS CLI, para que puedas usar el método que más te convenga. No hay diferencia entre los recursos de autorizadores personalizados creados por ninguno de los dos métodos.

Crear un recurso de autorización de clientes

Elija una de estas opciones para crear el recurso de autorizador personalizado

- [Elija un autorizador personalizado a través de la AWS IoT consola \(p. 248\)](#)
- [Elija un autorizador personalizado a través de la AWS CLI \(p. 249\)](#)

Para crear un autorizador personalizado (consola)

1. Abra el ícono [Página de autorización personalizada de AWS IoT consola](#), y elija **Crear**.
2. En **Creación de autorizador personalizado**:

- a. EnDale un nombre a su autorizador personalizado, introduzca **my-new-authorizer**.
- b. EnFunción de autorizador, elija la función Lambda que creó anteriormente.
- c. EnValidación de tokens: opcional:
 - i. ComprobarHabilitar firma de token.
 - ii. EnNombre de encabezado de token (opcional), introduzca **tokenKeyName**.
 - iii. EnNombre de la clave, introduzca **FirstKey**.
 - iv. EnValor, introduzca el contenido de la `public-key.pem`. Asegúrese de incluir las líneas del archivo con `-----BEGIN PUBLIC KEY-----` y `-----END PUBLIC KEY-----` no agregues ni elimines ninguna fuente de línea, devolución de carro u otros caracteres del contenido del archivo. La cadena que introduzca debe ser similar a lo que se muestra a continuación.

```
-----BEGIN PUBLIC KEY-----  
MIICIJANBgkqhkiG9w0BAQEFAOCAg8AMIIICCgKCAgEAvEBzOk4vhN+3Lgs1vEWt  
sLCqNmt5Damas3bmiTRvq2gjRJ6KXGTGQChqArAJwL1a9dkS9+maaXC3vc6xzx9z  
QPu/vQOe5tyzz1MsKdmtFGxMqQ3qjEXAMPLEOmqyUKPP5mff58k6ePSfXAnzBH0q  
lg2HioefrpU5OSAnpuRAjYKofKjbc2Vrn6N2G7hV+IfTBvCElf0csals/Rk4phD5  
oa4Y0GHISRnevypg5C8n9Rrz91PWGqP6M/q5DNJJXjMyleG92hQgu1N696bn5Dw8  
FhedszFa6b2x6xrItZFzewNQkPMLMFhNrQIIyvshT/F1LVCS5+v8AQ8UGGdfZmv  
QeqAMAF7WgagDMXcfgKSVU8yid2sIm56qsCLMvD2Sq8Lgzpey9N5ON1o1Cvldwvc  
KrJJtgwWhVqRGuShownLpgG86M6neZ5sRmbVNZ080zcobLngJOIbw9KkcUdklW  
gvZ6HEJqBY2XE70iEXAMPLETPHzhqvK6Ei1HGxpHsXx6BNft582J1VpgYjXha8o  
/NN7l7zbj/euAb41IVtmX8JrD9z613d1iM5L8HluJ1Uzn62Q+VeNV2tdA7MfpfMC  
8btGYladFAnitThaz6+F0VSBJPu7pZQoLnqyEp5zMtF+kFl2yOBmGAP0RBivRd9  
JWBUCG0bqcLQPeQyjbXSOfUCAwEAAQ==  
-----END PUBLIC KEY-----
```

3. ComprobarActivar autorizador.
4. ElegirCrear un autorizador.
5. Si se creó el recurso de autorización personalizado, verás la lista de autorizadores personalizados y tu nuevo autorizador personalizado debería aparecer en la lista y podrás continuar en la siguiente sección para probarlo.

Si ves un error, revisa el error e intenta volver a crear tu autorizador personalizado y vuelve a comprobar las entradas. Tenga en cuenta que cada recurso de autorizador personalizado debe tener un nombre único.

Para crear un autorizador personalizado (AWS CLI)

1. Sustituya los valores por `authorizer-function-arn`, `token-signing-public-keys`, a continuación, ejecute el siguiente 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-----  
MIICIJANBgkqhkiG9w0BAQEFAOCAg8AMIIICCgKCAgEAvEBzOk4vhN+3Lgs1vEWt  
sLCqNmt5Damas3bmiTRvq2gjRJ6KXGTGQChqArAJwL1a9dkS9+maaXC3vc6xzx9z  
QPu/vQOe5tyzz1MsKdmtFGxMqQ3qjEXAMPLEOmqyUKPP5mff58k6ePSfXAnzBH0q  
lg2HioefrpU5OSAnpuRAjYKofKjbc2Vrn6N2G7hV+IfTBvCElf0csals/Rk4phD5  
oa4Y0GHISRnevypg5C8n9Rrz91PWGqP6M/q5DNJJXjMyleG92hQgu1N696bn5Dw8  
FhedszFa6b2x6xrItZFzewNQkPMLMFhNrQIIyvshT/F1LVCS5+v8AQ8UGGdfZmv
```

```
OeqAMAF7WgagDMXcfgKSVU8yid2sIm56qsCLMvD2Sg8Lgzpey9N5ON1o1Cvldwvc
KrJJtgw6hVqRGuShownLpgG86M6neZ5sRMBVNZO80zcobLngJ01bw9KkcUdk1W
gvZ6HEJqBY2XE70iEXAMPLEPHzhqvK6Ei1HGxpHsXx6BNft582J1VpgYjXha8oa
/NN7l7zbj/euAb41IVtmX8JrD9z613d1iM5L8HluJLUzn62Q+VeNV2tdA7MfPfMC
8btGYladFAnitThaz6+F0VSBJPu7pZQoLnqyEp5zLMtF+kFl2yOBmGAP0RBivRd9
JWBUCG0bqcLQPeQyjbXSofUCAwEAAQ==
-----END PUBLIC KEY-----"
```

Donde:

- La `authorizer-function-arnvalue` es el nombre de recurso de Amazon (ARN) de la función de Lambda que creó para el autorizador personalizado.
- La `token-signing-public-keysvalor` incluye el nombre de la clave, `FirstKey`, y el contenido de `lapublic-key.pemfile`. Asegúrese de incluir las líneas del archivo con `-----BEGIN PUBLIC KEY-----` y `-----END PUBLIC KEY-----` y no agregues ni elimines ninguna fuente de línea, devolución de carro u otros caracteres del contenido del archivo.

Nota: tenga cuidado al introducir la clave pública, ya que cualquier modificación del valor de la clave pública la hace inutilizable.

2. Si se crea el autorizador personalizado, el comando devuelve el nombre y el ARN del nuevo recurso, como los siguientes.

```
{
    "authorizerName": "my-new-authorizer",
    "authorizerArn": "arn:aws:iot:Region:57EXAMPLE833:authorizer/my-new-authorizer"
}
```

Save the `authorizerArn` valor para su uso en el siguiente paso.

Recuerde que cada recurso de autorizador personalizado debe tener un nombre único.

Autorizar el recurso de autorizador personalizado

En esta sección, concederá permiso al recurso de autorización personalizado que acaba de crear permiso para ejecutar la función Lambda.

Conceda permiso a su función Lambda mediante el AWS CLI

1. Despues de insertar los valores, introduzca el siguiente comando. Tenga en cuenta que `statement-id` el valor debe ser único. Reemplazar `Id-1234` con otro valor si ha ejecutado este tutorial antes o si obtiene un `ResourceConflictException`.

```
aws lambda add-permission \
--function-name "custom-auth-function" \
--principal "iot.amazonaws.com" \
--action "lambda:InvokeFunction" \
--statement-id "Id-1234" \
--source-arn authorizerArn
```

2. Si el comando se ejecuta correctamente, devuelve una sentencia de permiso, como en este ejemplo. Puede continuar con la siguiente sección para probar el 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\"}}}"}
```

}

Si el comando no funciona correctamente, devuelve un error, como este ejemplo. Tendrás que revisar y corregir el error 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
```

Paso 4: Probar el autorizador llamando test-invoke-authorizer

Con todos los recursos definidos, en esta sección, llamarás a test-invoke-authorizer desde la línea de comandos para probar la aprobación de autorización.

Tenga en cuenta que cuando se invoca el autorizador desde la línea de comandos, protocolData.no está definido, por lo que el autorizador siempre devolverá un documento DENY. Sin embargo, esta prueba confirma que el autorizador personalizado y la función Lambda están configurados correctamente, incluso si no prueba completamente la función Lambda.

Para probar su autorizador personalizado y su función Lambda mediante el AWS CLI

1. En el directorio que contiene la `private-key.pem` archivo que creó en un paso anterior, ejecute el siguiente comando.

```
echo -n "tokenKeyValue" | openssl dgst -sha256 -sign private-key.pem | openssl base64 -A
```

Este comando crea una cadena de firma para utilizarla en el siguiente paso. La cadena de firma tiene un aspecto similar a este:

```
dBwykzlb+fo+JmSGdwoGr8dyC2qB/IyLefJJr+rbCvmu9Jl4KHA9DG+V  
+MMWu09YSA86+64Y3Gt4tOykpZqn9mn  
VB1wyxp+0bDZh8hmqUAUH3fwi3fPjBvCa4cwNuLQNgBZzbCvsluv7i2IMjEg  
+CPY0zrWt1jr9BikgGPdxWkjaaeh  
bQHHTo357TegKs9pP30Uf4TrxypNmFswA5k7QIC01n4bIyRTm900yZ94R4bdJsHNig1JePgnsOBvMGCEFE09jGjj  
szEHfgAUAQIWXiVGQj16BU1xKpTGSiTawheLKUjITOEXAMPLECK3aHKYKY  
+d1vTvdtKtYHBq8MjhzJ0kggbt29V  
QJCb8RilN/P5+vcVniSXWPlyB5jkYs9UvG08REoy64AtizfUhvSul/r/F3VV8ITtQp3aXiUtcspACi6ca  
+tsDuX  
f3LzCwQQF/YSuY02u5XkWn+sto6KCkpNlkD0wU8gl3+kOzxrthnQ8gEajd5Iylx230iqcXo3osjPha7JDyWM5o  
+K  
EWckTe911mokDr5sJ4JXixvnJTVSxlli49ialW4en1DAkc1a0s2U2UNm236EXAMPLElotyh7h  
+f1FeloZ1AWQFH  
xRlxSpqiVKS1ZIUClazWprh/orDJplpiWfBgBIogokJIDGP9gwhXIIk7zWrGmWpMK9o=
```

Copie esta cadena de firma para utilizarla en el siguiente paso. Tenga cuidado de no incluir ningún personaje adicional ni dejar de lado ninguno.

2. En este comando, sustituya la `token-signature` valor con la cadena de firma del paso anterior y ejecute este comando para probar el autorizador.

```
aws iot test-invoke-authorizer \  
--authorizer-name my-new-authorizer \  
--token tokenKeyValue \  
--token-signature dBwykzlb+fo+JmSGdwoGr8dyC2qB/IyLefJJr  
+rbCvmu9Jl4KHA9DG+V+MMWu09YSA86+64Y3Gt4tOykpZqn9mnVB1wyxp  
+0bDZh8hmqUAUH3fwi3fPjBvCa4cwNuLQNgBZzbCvsluv7i2IMjEg
```

```
+CPY0zrWt1jr9BikgGPDxWkjaaehbQHHTo357TegKs9pP30Uf4TrxyNmFswA5k7Q1c01n4bIyRTm90OyZ94R4bdJshNig1JePg
+d1vTvdtKtYHBq8MjhZ0kggbt29VQJCb8RilN/P5+vcVniSXWPplyB5jkYs9UvG08REoy64AtizfUhvSul/
r/F3VV8ITtQp3axiUtcpACi6ca+tsduXf3LzCwQOF/YSUY02u5XkWn
+sto6KCkpNlkD0wU8g13+kOzxrhthQ8gEajd5Iylx230iqcXo3osjPha7JDyWM5o
+KEWckTe91I1mokDr5sJ4JXixvnJTVSx1li49IalW4en1DAKc1a0s2U2UNm236EXAMPLELotyh7h
+f1FeloZlAWQFHxRlxSpqivKS1ZIUClazWprh/orDjplpiWfBgbIOgokJIDGP9gwhXIIk7zWrGmWpMK9o=
```

Si el comando se ejecuta correctamente, devuelve la información generada por la función de autorizador de clientes, como este ejemplo.

```
{
    "isAuthenticated": true,
    "principalId": "principalId",
    "policyDocuments": [
        "{\"Version\":\"2012-10-17\", \"Statement\":[{\"Action\":\"iot:*\", \"Effect\":
        \"Deny\", \"Resource\":\"arn:aws:iot:Region:57EXAMPLE833:*\"]}]"
    ],
    "refreshAfterInSeconds": 600,
    "disconnectAfterInSeconds": 3600
}
```

Si el comando devuelve un error, revise el error y compruebe dos veces los comandos que utilizó en esta sección.

Paso 5: Probar la publicación de mensajes MQTT con Postman

1. Para obtener el punto final de datos del dispositivo desde la línea de comandos, llame **describe-endpoint** como se muestra aquí

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

Guarde esta dirección para utilizarla como **device_data_endpoint_address** en un paso posterior.

2. Abra una nueva ventana de cartero y cree una nueva solicitud HTTP POST.
 - a. Abra la aplicación Postman desde su equipo.
 - b. En Postman, en el Archivo menú, elija Nuevo....
 - c. En el navegador Nuevo cuadro de diálogo, elija Solicitud.
 - d. En Save request,
 - i. En Nombre de la solicitud entrar **Custom authorizer test request**.
 - ii. En Seleccione una colección o carpeta en la que guardar: elija o cree una colección en la que guardar esta solicitud.
 - iii. Elegir Guardar en **nombre_colección_**.
3. Cree la solicitud POST para probar su autorizador personalizado.
 - a. En el selector de métodos de solicitud junto al campo URL, elija POST.
 - b. En el campo URL, cree la URL de su solicitud utilizando la siguiente URL con la **device_data_endpoint_address** desde las **describe-endpoint** en un paso anterior.

```
https://device_data_endpoint_address:443/topics/test/cust-auth/topic?
qos=0&actionToken=allow
```

Tenga en cuenta que esta URL incluye el actionToken=allow parámetro de consulta que indicará a la función Lambda que devuelva un documento de política que permita acceder

aAWS IoT. Después de introducir la URL, los parámetros de consulta también aparecen en elParamspestaña de Postman.

- c. En el navegadorAuth, en la pestañaTipocampo, elijaSin autenticación.
- d. En la pestaña Encabezados:
 - i. Si hay unAnfitriónclave que está marcada, desmarque esta.
 - ii. En la parte inferior de la lista de encabezados, agregue estos nuevos encabezados y confirme que están marcados. Sustituya elHostvalor con tu`device_data_endpoint_address`y `lax-amz-customauthorizer-signature`valor con la cadena de firma que usó con eltest-invoke-authorizedde la sección anterior.

Clave	Valor
<code>x-amz-customauthorizer-name</code>	<code>my-new-authorizer</code>
<code>Host</code>	<code>device_data_endpoint_address</code>
<code>tokenKeyName</code>	<code>tokenKeyValue</code>
<code>x-amz-customauthorizer-signature</code>	<code>DBWYKZLB+FO+JMSGDWGR8DYC2QB/ IYLEF JJR+RBCVMU9JL4 KHA9 DG+V+MMWU09YSA86+64Y3GT4 a YKPZQN9MNVB1 WYXP+0BDZH8HM QUAUH3FWI3FPJBVCA4CWNULQNQN BZZBCVS SLUV7I2IMJEG+CPY0ZRWT1JR9 BIKGPDWXKJAEEEHBOHTO357TEGKS9PP30UF4TRXYPNMF 90 OYZ94R4BDJSHNIG 1 JEPGNUOBVM GCEFE09 JJSZEHF AQI WXIV GQJ16 BU1 xkptgsita kujito de rueda ejemplo Pleck 3 AHKYKY +D1VTVDTHKYHBQ8MJHZJ0KGBT29VQJCB8RILN/ P5+VCVNISXWPPLYB5JKYS9UVG08REOY64 AIZFUHVSUL/R/F3VV8ITTQP3 AXI UTC Spaci 6CA+TSDUXF3LZCWQQF/ YSUY02U5xKWN+ST06 KCKPNLKDOwu8GL3+KOZXRTHNQ8GEAJD5IYLX230IQCXC +KEWCKTE91I1MOKDR5SJ4JXTV SX1LI49IALW4EN1DAKC1A0S2U2UNM236 Ejemplo LOTY H7 H+FLFFLAW QFHXRXLXSPQIVKS1ZIUCLAZWPRA/ ORDJPLPIWFGBIOGOKJIDGP9GWHXIIIK7ZWRGMWPMK9C</code>

- e. En la pestaña Cuerpo:
 - i. En el cuadro de opciones de formato de datos, elijaRaw.
 - ii. En la lista de tipos de datos, elijaJavaScript.
 - iii. En el campo de texto, introduzca esta carga útil de mensaje JSON para el mensaje de prueba:

```
{  
    "data_mode": "test",  
    "vibration": 200,  
    "temperature": 40  
}
```

4. ElegirSendpara enviar la solicitud.

Si la solicitud se realizó correctamente, devuelve:

```
{  
    "message": "OK",  
    "traceId": "ff35c33f-409a-ea90-b06f-fbEXAMPLE25c"  
}
```

La respuesta correcta indica que el autorizador personalizado ha permitido la conexión aAWS IoT que el mensaje de prueba se entregó al corredor enAWS IoT Core.

Si devuelve un error, revise el mensaje de error,[device_data_endpoint_address](#), la cadena de firma y los demás valores de encabezado.

Conserve esta solicitud en Postman para utilizarla en la siguiente sección.

Paso 6: Ver mensajes en el cliente de prueba MQTT

En el paso anterior, envió mensajes de dispositivo simulados aAWS IoTUtilizando Postman. La respuesta correcta indica que el autorizador personalizado permitió la conexión aAWS IoT que el mensaje de prueba se entregó al corredor enAWS IoT Core. En esta sección, utilizará el cliente de prueba MQTT en laAWS IoTconsola para ver el contenido del mensaje de ese mensaje como podrían hacerlo otros dispositivos y servicios.

Para ver los mensajes de prueba autorizados por su autorizador personalizado

1. En el navegadorAWS IoTConsola de, abra[Cliente de pruebas MQTT](#).
2. En el navegadorSuscripción al temapestaña, enFiltro de temas, introduzca**test/cust-auth/topic**, que es el tema del mensaje utilizado en el ejemplo de Cartero de la sección anterior.
3. Elija Subscribe.

Mantenga visible esta ventana para el siguiente paso.

4. En Postman, en la solicitud que creó para la sección anterior, elijaSend.

Revise la respuesta para asegurarse de que se ha realizado correctamente. Si no es así, solucione el error como se describe en la sección anterior.

5. En el navegadorCliente de pruebas MQTT, debería ver una nueva entrada que muestre el tema del mensaje y, si se expande, la carga útil del mensaje de la solicitud que envió desde Postman.

Si los mensajes no aparecen en laCliente de pruebas MQTT, estas son algunas cosas que se deben comprobar:

- Asegúrate de que tu solicitud de cartero se haya devuelto correctamente. SiAWS IoTchaza la conexión y devuelve un error, el mensaje de la solicitud no se pasa al agente de mensajes.
- Asegúrese de queCuenta de AWSyRegión de AWSutilizado para abrirAWS IoTla consola es la misma que usas en la URL de cartero.
- Asegúrate de haber introducido el tema correctamente en elCliente de pruebas MQTT. El filtro de temas distingue entre mayúsculas y minúsculas En caso de duda, también puede suscribirse al#, que se suscribe a todos los mensajes MQTT que pasan por el agente de mensajesCuenta de AWSyRegión de AWSutilizado para abrirAWS IoTconsola de .

Paso 7: Revisar los resultados y los próximos pasos

En este tutorial:

- Ha creado una función Lambda para ser un gestor de autorizador personalizado
- Ha creado un autorizador personalizado con la firma de tokens habilitada
- Probaste tu autorizador personalizado utilizando el comando `test-invoke-authorizer`
- Publicó un tema MQTT mediante [Postman](#) valide la solicitud con su autorizador personalizado
- Usaste el Cliente de pruebas MQTT para ver los mensajes enviados desde la prueba de cartero

Pasos siguientes

Después de enviar algunos mensajes de Postman para verificar que el autorizador personalizado está funcionando, intente experimentar para ver cómo los cambios en los distintos aspectos de este tutorial afectan a los resultados. A continuación se muestran algunos ejemplos para comenzar.

- Cambie la cadena de firma para que ya no sea válida para ver cómo se gestionan los intentos de conexión no autorizados. Debería recibir una respuesta de error, como esta, y el mensaje no debe aparecer en el Cliente de pruebas MQTT.

```
{  
  "message": "Forbidden",  
  "traceId": "15969756-a4a4-917c-b47a-5433e25b1356"  
}
```

- Para obtener más información sobre cómo encontrar errores que podrían producirse durante el desarrollo y el uso AWS IoT, consulte [Monitorización de AWS IoT \(p. 426\)](#).

Paso 8: Eliminar recursos

Si quieras repetir este tutorial, es posible que tengas que eliminar algunos de tus autorizadores personalizados. Una cuenta de AWS puede tener un número limitado de autorizadores personalizados configurados a la vez y puede obtener una `LimitExceededException` cuando intentas añadir uno nuevo sin quitar un autorizador personalizado existente.

Para quitar un autorizador personalizado (consola)

1. Abra el icono [Página de autorización personalizada del AWS IoT](#) consola, y en la lista de autorizadores personalizados, busque el autorizador personalizado que desea eliminar.
2. Abra la página de detalles del autorizador personalizado y, desde el menú Actions, elija Editar.
3. Desmarque la casilla Activar autorizador y luego elija Actualización.
No se puede eliminar un autorizador personalizado mientras está activo.
4. En la página de detalles del autorizador personalizado, abra el menú Actions y elija Borrar.

Para eliminar un autorizador personalizado (AWS CLI)

1. Enumere los autorizadores personalizados que ha instalado y busque el nombre del autorizador personalizado que desea eliminar.

```
aws iot list-authors
```

2. Defina el autorizador personalizado en `inactive` ejecutando este comando después de sustituir `Custom_Auth_Name` con `authorizerName` del autorizador personalizado que se eliminará.

```
aws iot update-authorizer --status INACTIVE --authorizer-name Custom_Auth_Name
```

3. Elimine el autorizador personalizado ejecutando este comando después de reemplazar *Custom_Auth_Name* con `authorizerName` del autorizador personalizado que se eliminará.

```
aws iot delete-authorizer --authorizer-name Custom_Auth_Name
```

Tutorial: Control de la humedad del suelo conAWS IoT Raspberry Pi

En este tutorial se muestra cómo utilizar un [Raspberry Pi](#), un sensor de humedad, e AWS IoT para monitorizar el nivel de humedad del suelo de una planta o un terreno. El Raspberry Pi ejecuta código que lee el nivel de humedad y la temperatura del sensor y, a continuación, envía los datos a AWS IoT. Cree una regla en AWS IoT que envía un correo electrónico a una dirección suscrita a un tema de Amazon SNS cuando el nivel de humedad está por debajo de un umbral.

Note

Es posible que este tutorial no esté actualizado. Es posible que algunas referencias se hayan sustituido desde que se publicó originalmente este tema.

Contenido

- [Requisitos previos \(p. 256\)](#)
- [Configuración de AWS IoT \(p. 256\)](#)
 - [Paso 1: Cree la política de AWS IoT. \(p. 257\)](#)
 - [Paso 2: Creación de la clave privada, certificado y objeto de AWS IoT \(p. 258\)](#)
 - [Paso 3: Crear un tema de Amazon SNS. \(p. 259\)](#)
 - [Paso 4: Creación de una regla de AWS IoT para enviar un correo electrónico \(p. 259\)](#)
- [Configuración del dispositivo Raspberry Pi y el sensor de humedad \(p. 260\)](#)

Requisitos previos

Para completar este tutorial, se necesita lo siguiente:

- Una Cuenta de AWS.
- Un usuario de IAM con permisos de administrador.
- Un equipo de desarrollo con Windows, macOS, Linux o Unix para obtener acceso a la [consola de AWS IoT](#).
- UNA [Raspberry Pi 3B o 4B](#) ejecutando lo último [Sistema operativo Raspbian](#). Para ver instrucciones de instalación, consulte [Instalación de imágenes del sistema operativo](#) en el sitio web de Raspberry Pi.
- Un monitor, teclado, ratón y conexión de red wifi o Ethernet para su Raspberry Pi.
- Un sensor de humedad compatible con Raspberry Pi. El sensor utilizado en este tutorial es un [sensor de humedad capacitivo Adafruit STEMMA I2C](#) con un conector hembra de 4 clavijas JST.

Configuración de AWS IoT

Para completar este tutorial, debe crear los siguientes recursos. Para conectar un dispositivo a AWS IoT, debe crear un objeto de IoT, un certificado de dispositivo y una política de AWS IoT.

- Un objeto de AWS IoT.

Un objeto representa un dispositivo físico (en este caso, su Raspberry Pi) e incluye metadatos estáticos sobre el dispositivo.

- Un certificado de dispositivo.

Todos los dispositivos deben tener un certificado de dispositivo para conectarse a AWS IoT y autenticarse con el mismo.

- Una política de AWS IoT.

Cada certificado de dispositivo tiene una o varias políticas de AWS IoT asociadas a él. Estas políticas determinan a qué recursos de AWS IoT puede obtener acceso el dispositivo.

- Un certificado de CA raíz de AWS IoT.

Los dispositivos y otros clientes utilizan un certificado de CA raíz de AWS IoT para autenticar el servidor de AWS IoT con el que se están comunicando. Para obtener más información, consulte [Autenticación del servidor \(p. 295\)](#).

- Una regla de AWS IoT.

Una regla incluye una consulta y una o varias acciones de regla. La consulta extrae datos de los mensajes del dispositivo para determinar si los datos de los mensajes deben procesarse. La acción de regla especifica qué hacer si los datos coinciden con la consulta.

- Un tema de Amazon SNS.

La regla escucha los datos de humedad del dispositivo Raspberry Pi. Si el valor está por debajo de un umbral, envía un mensaje al tema de Amazon SNS. Amazon SNS envía dicho mensaje a todas las direcciones de correo electrónico suscritas al tema.

Paso 1: Cree la política de AWS IoT.

Cree una política de AWS IoT que permita al dispositivo Raspberry Pi conectarse y enviar mensajes a AWS IoT.

1. En la [consola de AWS IoT](#), si aparece un botón Get started (Empezar), elíjalo. De lo contrario, en el panel de navegación, expanda Secure (Seguridad) y, a continuación, elija Policies (Políticas).
2. Si aparece el cuadro de diálogo You don't have any policies yet (Aún no tiene ninguna política), elija Create a policy (Crear una política). De lo contrario, seleccione Create.
3. Escriba un nombre para la política de AWS IoT (por ejemplo, **MoistureSensorPolicy**).
4. En la sección Add statements (Añadir instrucciones), sustituya la política existente por el siguiente JSON. Reemplazar **region** y **cuenta** con el Región de AWS y Cuenta de AWS número.

```
{  
    "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. Elija Create (Crear).

Paso 2: Creación de la clave privada, certificado y objeto de AWS IoT

Cree un objeto en el registro de AWS IoT para representar su Raspberry Pi.

1. En la [consola de AWS IoT](#), en el panel de navegación, elija Manage (Administrar) y, a continuación, seleccione Things (Objetos).
2. Si aparece el cuadro de diálogo You don't have any things yet (Aún no tiene ningún objeto), elija Register a thing (Registrar un objeto). De lo contrario, seleccione Create.
3. En la página Creación de objetos de AWS IoT, elija Crear un solo objeto.
4. En la página Add your device to the device registry (Añadir su dispositivo al registro de dispositivos), escriba un nombre para el objeto de IoT (por ejemplo, **RaspberryPi**) y, a continuación, elija Next (Siguiente). No puede modificar el nombre de un objeto una vez creado. Para cambiar el nombre de un objeto, debe crear otro objeto nuevo, asignarle el nuevo nombre y eliminar después el objeto anterior.

5. En la página Add a certificate for your thing (Aregar un certificado para el objeto), elija Create certificate (Crear certificado).
6. Elija los enlaces Download (Descargar) para descargar el certificado, la clave privada y el certificado de CA raíz.

Important

Esta es la única vez que puede descargar el certificado y la clave privada.

7. Para activar el certificado, elija Activar. El certificado debe estar activo para que un dispositivo se conecte a AWS IoT.
8. Elija Asociar una política.
9. En Add a policy for your thing (Añadir una política al objeto), elija MoistureSensorPolicy y, a continuación, seleccione Register Thing (Registrar objeto).

Paso 3: Crear un tema de Amazon SNS.

Cree un tema de Amazon SNS.

1. Desde el [AWS consola SNS](#), en el panel de navegación, seleccione Temas y luego seleccione Crear tema.
2. Escriba un nombre para el tema (por ejemplo, **MoistureSensorTopic**).
3. Escriba un nombre de visualización para el tema (por ejemplo, **Moisture Sensor Topic**). Este es el nombre que se muestra para el tema en la consola de Amazon SNS.
4. Elija Create new topic.
5. En la página de detalles del tema de Amazon SNS, seleccione Crear suscripción.
6. En Protocol (Protocolo), elija Email (Correo electrónico).
7. En Punto de enlace, introduzca su dirección de correo electrónico.
8. Elija Create subscription (Crear suscripción).
9. Abra su cliente de correo electrónico y busque un mensaje con el asunto **MoistureSensorTopic**. Abra el correo electrónico y haga clic en el enlace Confirm subscription (Confirmar suscripción).

Important

No recibirá ninguna alerta por correo electrónico de este tema de Amazon SNS hasta que confirme la suscripción.

Debería recibir un mensaje de correo electrónico con el texto que escribió.

Paso 4: Creación de una regla de AWS IoT para enviar un correo electrónico

Una regla de AWS IoT define una consulta y una o varias acciones que se deben realizar cuando se recibe un mensaje de un dispositivo. El motor de reglas de AWS IoT escucha los mensajes enviados por los dispositivos y utiliza los datos de los mensajes para determinar si se debe realizar alguna acción. Para obtener más información, consulte [Reglas para AWS IoT \(p. 472\)](#).

En este tutorial el dispositivo Raspberry Pi publica mensajes en `aws/things/RaspberryPi/shadow/update`. Se trata de un tema de MQTT interno utilizado por los dispositivos y el servicio Thing Shadow. El Raspberry Pi publica mensajes que tienen el siguiente formato:

```
{  
  "reported": {  
    "moisture" : moisture-reading,  
    "temp" : temperature-reading  
  }  
}
```

}

Puede crear una consulta que extraiga los datos de humedad y temperatura del mensaje entrante. También puede crear una acción de Amazon SNS que tome los datos y los envíe a los suscriptores del tema de Amazon SNS si la lectura de humedad está por debajo de un valor de umbral.

Crear una regla de Amazon SNS

1. En la [consola de AWS IoT](#), en el panel de navegación, elija Act (Actuar). Si aparece el cuadro de diálogo You don't have any rules yet (Aún no tiene ninguna regla), elija Create a rule (Crear una regla). De lo contrario, seleccione Create.
2. En la página Create a rule (Crear una regla), introduzca un nombre para la regla (por ejemplo, **MoistureSensorRule**).
3. En Description (Descripción), proporcione una breve descripción de esta regla, (por ejemplo, **Sends an alert when soil moisture level readings are too low**).
4. En Rule query statement (Instrucción de consulta de regla), elija SQL versión 2016-03-23 e introduzca la siguiente instrucción de consulta SQL de AWS IoT:

```
SELECT * FROM '$aws/things/RaspberryPi/shadow/update/accepted' WHERE state.reported.moisture < 400
```

Esta instrucción activa la acción de la regla cuando la lectura de `moisture` es menor que 400.

Note

Es posible que tenga que utilizar un valor diferente. Una vez que el código se ejecute en el dispositivo Raspberry Pi, si toca el sensor, lo coloca en agua o en una maceta, podrá ver los valores que se obtienen del sensor.

5. En Set one or more actions (Definir una o varias acciones), elija Add action (Añadir acción).
6. En la página Seleccionar una acción, elija Enviar un mensaje como una notificación push SNS.
7. Desplácese hasta la parte inferior de la página y, a continuación, elija Configure action (Configurar acción).
8. En la página Configure action (Configurar acción), en SNS target (Objetivo de SNS), elija Select (Seleccionar) y, a continuación, seleccione LowMoistureTopic.
9. En Formato del mensaje, elija RAW.
10. UNDERElegir o crear un rol para conceder AWS IoT acceso para realizar esta acción, elige Crear rol. Especifique un nombre para el rol (por ejemplo, **LowMoistureTopicRole**) y elija Create role (Crear rol).
11. Elija Añadir acción.
12. Elija Create rule (Crear regla).

Configuración del dispositivo Raspberry Pi y el sensor de humedad

Inserte la tarjeta microSD en el dispositivo Raspberry Pi, conecte el monitor, el teclado, el ratón y, si no utiliza Wi-Fi, el cable Ethernet. No conecte aún el cable de alimentación.

Conecte el cable puente JST al sensor de humedad. El otro lado del puente tiene cuatro cables:

- Green (Verde): I2C SCL

- Blanco: I2C SDA
- Rojo: alimentación (3,5 V)
- Negro: conexión a tierra

Sujete el dispositivo Raspberry Pi con el enchufe hembra Ethernet a la derecha. En esta orientación hay dos filas de clavijas GPIO en la parte superior. Conecte los cables del sensor de humedad a la fila inferior de clavijas en el orden que se indica a continuación. Comenzando por la clavija del extremo izquierdo, conecte el cable rojo (alimentación), el cable blanco (SDA) y el cable verde (SCL). Omite una clavija y, a continuación, conecte el cable negro (conexión a tierra). Para obtener más información, consulte [Cableado de equipos Python](#).

Conecte el cable de alimentación al dispositivo Raspberry Pi y conecte el otro extremo a una toma de corriente para encenderlo.

Configuración del dispositivo Raspberry Pi

1. En Welcome to Raspberry Pi (Bienvenido a Raspberry Pi), elija Next (Siguiente).
2. Elija el país, el idioma, la zona horaria y la distribución del teclado. Elija Next (Siguiente).
3. Escriba una contraseña para el dispositivo Raspberry Pi y, a continuación, elija Next (Siguiente).
4. Elija una red wifi y, a continuación, elija Next (Siguiente). Si no utiliza una red wifi, elija Skip (Omitir).
5. Elija Next (Siguiente) para comprobar si hay actualizaciones de software. Cuando se completen las actualizaciones, elija Restart (Reiniciar) para reiniciar el dispositivo Raspberry Pi.

Una vez que se inicie el dispositivo Raspberry Pi, habilite la interfaz de I2C.

1. En la esquina superior izquierda del escritorio de Raspbian, haga clic en el icono de Raspberry, elija Preferences (Preferencias) y, a continuación, elija Raspberry Pi Configuration (Configuración de Raspberry Pi).
2. En la pestaña Interfaces (Interfaces), en I2C, elija Enable (Habilitar).
3. Seleccione OK (Aceptar).

Las bibliotecas del sensor de humedad Adafruit STEMMA se han escrito para CircuitPython. Para ejecutarlas en un dispositivo Raspberry Pi, debe instalar la última versión de Python 3.

1. Ejecute los siguientes comandos desde un símbolo del sistema para actualizar el software del dispositivo Raspberry Pi:

```
sudo apt-get update
```

```
sudo apt-get upgrade
```

2. Ejecute el siguiente comando para actualizar la instalación de Python 3:

```
sudo pip3 install --upgrade setuptools
```

3. Ejecute el siguiente comando para instalar las bibliotecas de GPIO de Raspberry Pi:

```
pip3 install RPI.GPIO
```

4. Ejecute el siguiente comando para instalar las bibliotecas de Adafruit Blinka:

```
pip3 install adafruit-blinka
```

Para obtener más información, consulte la sección [Installing CircuitPython Libraries on Raspberry Pi](#).

5. Ejecute el siguiente comando para instalar las bibliotecas de Adafruit Seesaw:

```
sudo pip3 install adafruit-circuitpython-seesaw
```

6. Ejecute el siguiente comando para instalar el SDK de dispositivos AWS IoT para Python:

```
pip3 install AWSIoTPythonSDK
```

El dispositivo Raspberry Pi ahora tiene todas las bibliotecas necesarias. Cree un archivo denominado **moistureSensor.py** y copie el siguiente código Python en el archivo:

```
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)
```

Guarde el archivo en un lugar donde pueda encontrarlo. Ejecute `moistureSensor.py` desde la línea de comandos con los siguientes parámetros:

punto de conexión

Su punto de enlace de AWS IoT personalizado. Para obtener más información, consulte [API REST de sombra de dispositivo \(p. 653\)](#).

rootCA

El camino completo a las AWS IoT certificado de CA raíz.

cert

La ruta completa al certificado de dispositivos de AWS IoT.

key

La ruta completa a la clave privada del certificado de dispositivos de AWS IoT.

thingName

El nombre del objeto (en este caso, `RaspberryPi`).

clientId

El ID de cliente de MQTT. Use `RaspberryPi`.

La línea de comandos debería tener este aspecto:

```
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
```

Pruebe a tocar el sensor, colocarlo en una maceta o ponerlo en un vaso de agua para ver cómo responde a distintos niveles de humedad. Si es necesario, puede cambiar el valor del umbral en la opción `MoistureSensorRule`. Cuando la lectura del sensor de humedad está por debajo del valor especificado

en la instrucción de consulta SQL de la regla, AWS IoT publica un mensaje en el tema de Amazon SNS. Debería recibir un mensaje de correo electrónico que incluya los datos de humedad y temperatura.

Una vez que haya verificado la recepción de mensajes de correo electrónico de Amazon SNS, pulse CTRL + C para detener el programa Python. Es poco probable que el programa Python envíe suficientes mensajes para incurrir en gastos, pero se recomienda detener el programa cuando haya terminado.

Administración de dispositivos con AWS IoT

AWS IoT proporciona un registro que le ayuda a administrar los objetos. Un objeto es una representación de un dispositivo concreto o de una entidad lógica. Puede ser un dispositivo físico o un sensor (por ejemplo, una bombilla o un interruptor en la pared). También puede ser una entidad lógica, como una instancia de una aplicación o una entidad física que no se conecta con AWS IoT, pero que está relacionada con otros dispositivos que sí lo están (por ejemplo, un automóvil con sensores de motor o un panel de control).

La información sobre un objeto se almacena en el registro en forma de datos JSON. A continuación, se muestra un ejemplo de objeto:

```
{  
    "version": 3,  
    "thingName": "MyLightBulb",  
    "defaultClientId": "MyLightBulb",  
    "thingTypeName": "LightBulb",  
    "attributes": {  
        "model": "123",  
        "wattage": "75"  
    }  
}
```

Los objetos se identifican por su nombre. También pueden tener atributos, que son pares nombre-valor que puede utilizar para almacenar información acerca del objeto, como su número de serie o su fabricante.

En un caso de uso de dispositivo típico, se utiliza el nombre del objeto como ID de cliente MQTT predeterminado. Aunque no es obligatorio establecer un mapeo entre el nombre de un objeto en el registro y su uso de los ID de cliente MQTT, los certificados o el estado de sombra, le recomendamos elegir un nombre de objeto y utilizarlo como ID de cliente MQTT, tanto para el registro como para el servicio Device Shadow. De esta forma, puede organizar su flota de IoT más fácilmente, sin perder la flexibilidad del modelo de certificado de dispositivo subyacente o las sombras.

No es necesario crear un objeto en el registro para conectar un dispositivo a AWS IoT. Al añadir objetos al registro, podrá administrar y buscar dispositivos con más facilidad.

Cómo administrar objetos con el registro

Usa la AWS IoT consola, AWS IoT API o la AWS CLI para interactuar con el registro. En las secciones siguientes se muestra cómo utilizar la CLI para trabajar con el registro.

Al nombrar objetos de tu cosa:

- No debe utilizar datos personales en los nombres de objeto. El nombre de la cosa puede aparecer en comunicaciones e informes sin cifrar.
- No debe usar un carácter de dos puntos (:) en el nombre de una cosa. El carácter de dos puntos se utiliza como delimitador por otros AWS IoT y esto puede provocar que analicen cadenas con nombres de cosas incorrectamente.

Creación de un objeto

A continuación, se muestra cómo se utiliza el comando `CreateThing` de AWS IoT en la CLI para crear un objeto: No se puede modificar el nombre de un objeto una vez creado. Para cambiar el nombre de un objeto, debe crear otro objeto nueva, asignarle el nuevo nombre y eliminar después el objeto anterior.

```
$ aws iot create-thing --thing-name "MyLightBulb" --attribute-payload "{\"attributes\":{\"wattage\":\"75\", \"model\":\"123\"}}"
```

El comando `CreateThing` muestra el nombre y el ARN (nombre de recurso de Amazon) del nuevo objeto:

```
{
  "thingArn": "arn:aws:iot:us-east-1:123456789012:thing/MyLightBulb",
  "thingName": "MyLightBulb",
  "thingId": "12345678abcdefghijklmnopqrstuvwxyz"
}
```

Note

No es recomendable utilizar datos personales en los nombres de objeto.

Lista de objetos

Puede utilizar el comando `ListThings` para enumerar todos los objetos en su cuenta:

```
$ aws iot list-things
```

```
{
  "things": [
    {
      "attributes": {
        "model": "123",
        "wattage": "75"
      },
      "version": 1,
      "thingName": "MyLightBulb"
    },
    {
      "attributes": {
        "numOfStates": "3"
      },
      "version": 11,
      "thingName": "MyWallSwitch"
    }
  ]
}
```

Puede utilizar el `ListThings` para buscar todos los objetos de un tipo de objeto 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"
    }
]
```

Puede utilizar el `list-things` para buscar todos los objetos que tienen un atributo con un valor específico. Este comando busca solo los atributos que se pueden buscar.

```
$ 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"
        }
    ]
}
```

Si [indexación de flotas](#) (p. 825) está habilitado, puede utilizar el `search-index` para buscar atributos de cosa que se pueden buscar y no se pueden realizar búsquedas, valores de sombra de dispositivo y valores de conectividad. Para obtener más información acerca de lo que puede consultar mediante `search-index` comando, consulte [Ejemplo de consultas de objetos](#) (p. 847) y la referencia de la CLI sobre `search-index` comando.

Describe objetos

Puede utilizar el comando `DescribeThing` para mostrar información más detallada acerca de un objeto:

```
$ aws iot describe-thing --thing-name "MyLightBulb"
{
    "version": 3,
    "thingName": "MyLightBulb",
    "thingArn": "arn:aws:iot:us-east-1:123456789012:thing/MyLightBulb",
    "thingId": "12345678abcdefghijklmnopqrstuvwxyz",
    "defaultClientId": "MyLightBulb",
    "thingTypeName": "StopLight",
    "attributes": {
        "model": "123",
        "wattage": "75"
    }
}
```

Actualización de un objeto

Puede utilizar el comando `UpdateThing` para actualizar un objeto. Tenga en cuenta que este comando solo actualiza los atributos del objeto. El nombre de un objeto no se puede modificar. Para cambiar el nombre de un objeto, debe crear otro objeto nuevo, asignarle el nuevo nombre y eliminar después el objeto anterior.

```
$ aws iot update-thing --thing-name "MyLightBulb" --attribute-payload "{\"attributes\": {\"wattage\":\"150\", \"model\":\"456\"}}"
```

El comando `UpdateThing` no genera una salida. Puede utilizar el comando de `DescribeThing` para ver el resultado:

```
$ aws iot describe-thing --thing-name "MyLightBulb"
{
    "attributes": {
        "model": "456",
        "wattage": "150"
    },
    "version": 2,
    "thingName": "MyLightBulb"
}
```

Eliminación de un objeto

Puede utilizar el comando `DeleteThing` para eliminar un objeto:

```
$ aws iot delete-thing --thing-name "MyThing"
```

Este comando se devuelve correctamente sin error si la eliminación se realiza correctamente o especifica un objeto que no existe.

Asociar un principal a un objeto

Un dispositivo físico debe tener un certificado X.509 para comunicarse con AWS IoT. Puede asociar el certificado de un dispositivo con el objeto del registro que representa a dicho dispositivo. Para adjuntar un certificado a su objeto, utilice el comando `AttachThingPrincipal`:

```
$ aws iot attach-thing-principal --thing-name "MyLightBulb" --principal "arn:aws:iot:us-east-1:123456789012:cert/a0c01f5835079de0a7514643d68ef8414ab739a1e94ee4162977b02b12842847"
```

El comando AttachThingPrincipal no genera una salida.

Desvincular un principal de un objeto

Puede utilizar el comando DetachThingPrincipal para desvincular un certificado de un objeto:

```
$ aws iot detach-thing-principal --thing-name "MyLightBulb" --principal "arn:aws:iot:us-east-1:123456789012:cert/a0c01f5835079de0a7514643d68ef8414ab739a1e94ee4162977b02b12842847"
```

La comando DetachThingPrincipal no genera ningún resultado.

Tipos de objeto

Los tipos de objeto le permiten almacenar información descriptiva y de configuración común a todos los objetos asociados al mismo tipo de objeto. Esto simplifica la administración de objetos en el registro. Por ejemplo, puede definir un tipo de objeto LightBulb (bombilla). Todos los objetos asociados al tipo de objeto LightBulb comparten un conjunto de atributos: número de serie, fabricante y potencia. Al crear un objeto de tipo LightBulb (o cambiar el tipo a LightBulb) puede especificar valores para cada uno de los atributos definidos en este tipo de objeto.

Aunque los tipos de objeto son opcionales, su uso facilita la detección de objetos.

- Los objetos con un tipo de objeto pueden tener un máximo de 50 atributos.
- Los objetos sin tipo de objeto pueden tener un máximo de tres atributos.
- Los objetos solo se pueden asociar a un tipo de objeto.
- El número de tipos de objetos que puede crear en su cuenta es ilimitado.

Los tipos de objeto son inmutables. No se puede cambiar el nombre de un tipo de objeto después de que se haya creado. Puede descartar un tipo de objeto, en cualquier momento, para evitar que se le asocien nuevos objetos. También puede eliminar tipos de objeto que no tengan objetos asociados.

Creación de un tipo de objeto

Puede utilizar el comando CreateThingType para crear un tipo de objeto:

```
$ aws iot create-thing-type  
  --thing-type-name "LightBulb" --thing-type-properties  
  "thingTypeDescription=light bulb type, searchableAttributes=wattage,model"
```

El comando CreateThingType devuelve una respuesta que contiene el tipo de objeto y su ARN:

```
{  
  "thingTypeName": "LightBulb",  
  "thingTypeId": "df9c2d8c-894d-46a9-8192-9068d01b2886",  
  "thingTypeArn": "arn:aws:iot:us-west-2:123456789012:thingtype/LightBulb"  
}
```

Listado de los tipos de objeto

Puede utilizar el comando `ListThingTypes` para crear una lista de los tipos de objeto:

```
$ aws iot list-thing-types
```

La `ListThingTypes` devuelve una lista de los tipos de objeto definidos en su cuenta de AWS:

```
{
  "thingTypes": [
    {
      "thingTypeName": "LightBulb",
      "thingTypeProperties": {
        "searchableAttributes": [
          "wattage",
          "model"
        ],
        "thingTypeDescription": "light bulb type"
      },
      "thingTypeMetadata": {
        "deprecated": false,
        "creationDate": 1468423800950
      }
    }
  ]
}
```

Descripción de un tipo de objeto

Puede utilizar el comando `DescribeThingType` para obtener información acerca de un tipo de objeto:

```
$ aws iot describe-thing-type --thing-type-name "LightBulb"
```

El comando `DescribeThingType` devuelve información acerca del 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
  }
}
```

Asociación de un tipo de objeto a un objeto

Puede utilizar el comando `CreateThing` para especificar un tipo de objeto cuando crea un objeto:

```
$ aws iot create-thing --thing-name "MyLightBulb" --thing-type-name "LightBulb" --attribute-payload "{\"attributes\":{\"wattage\":\"75\",\"model\":\"123\"}}"
```

Puede utilizar el comando `UpdateThing` en cualquier momento para cambiar el tipo de objeto asociado a un objeto:

```
$ aws iot update-thing --thing-name "MyLightBulb"  
    --thing-type-name "LightBulb" --attribute-payload "{\"attributes\"::  
    {"wattage": "75", "model": "123"}}
```

También puede utilizar el comando `UpdateThing` para desvincular un objeto de un tipo de objeto.

Descartar un tipo de objeto

Los tipos de objeto son inmutables. No se pueden cambiar una vez que están definidos. Sin embargo, puede descartar un tipo de objeto para evitar que los usuarios le asocien nuevos objetos. Todos los objetos que estén asociados al tipo de objeto permanecen igual, sin cambios.

Para descartar un tipo de objeto, utilice el comando `DeprecateThingType`:

```
$ aws iot deprecate-thing-type --thing-type-name "myThingType"
```

Puede utilizar el comando de `DescribeThingType` para ver el 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  
    }  
}
```

Descartar un tipo de objeto es una operación reversible. Puede anular un descarte utilizando la marca `--undo-deprecate` con el comando de la CLI `DeprecateThingType`:

```
$ aws iot deprecate-thing-type --thing-type-name "myThingType" --undo-deprecate
```

Puede utilizar el comando de la CLI `DescribeThingType` para ver el resultado:

```
$ aws iot describe-thing-type --thing-type-name "StopLight":
```

```
{  
    "thingTypeName": "StopLight",  
    "thingTypeArn": "arn:aws:iot:us-east-1:123456789012:thingtype/StopLight",  
    "thingTypeId": "12345678abcdefg12345678ijklmnop12345678",  
    "thingTypeProperties": {  
        "searchableAttributes": [  
            "wattage",  
            "numOfLights",  
        ]  
    }  
}
```

```
        "model"
    ],
    "thingTypeDescription": "traffic light type"
},
"thingTypeMetadata": {
    "deprecated": false,
    "creationDate": 1468425854308,
}
}
```

Eliminación de un tipo de objeto

Puede eliminar tipos de objeto solo después de que se hayan descartado. Para eliminar un tipo de objeto, utilice el comando `DeleteThingType`:

```
$ aws iot delete-thing-type --thing-type-name "StopLight"
```

Note

Debe esperar cinco minutos después de descartar un tipo de objeto para poder eliminarlo.

Grupos de objetos estáticos

Los grupos de objetos estáticos permiten administrar varios objetos a la vez clasificándolos en grupos. Los grupos de objetos estáticos contienen un grupo de objetos que se administran a través de la consola, la CLI o la API. Por otro lado, los [grupos de objetos dinámicos \(p. 282\)](#) contienen objetos que coinciden con una consulta especificada. Los grupos de objetos estáticos también pueden contener otros grupos de objetos estáticos; puede crear una jerarquía de grupos. Puede asociar una política a un grupo principal y sus grupos secundarios la heredarán, así como todos los objetos del grupo y los grupos secundarios. Esto facilita el control de los permisos para un gran número de objetos.

Esto es lo que puede hacer con los grupos de objetos estáticos:

- Crear, describir o eliminar un grupo.
- Añadir un objeto a un grupo o a más de un grupo.
- Quitar un objeto de un grupo.
- Enumerar los grupos que ha creado.
- Enumerar todos los grupos secundarios de un grupo (sus descendientes directos e indirectos).
- Enumerar los objetos de un grupo, incluidos todos los objetos en sus grupos secundarios.
- Enumerar todos los grupos antecesores de un grupo (sus grupos principales directos e indirectos).
- Añadir, eliminar o actualizar los atributos de un grupo. (Los atributos son pares nombre-valor que puede usar para almacenar información acerca de un grupo).
- Adjuntar una política a un grupo o separarla de él.
- Enumerar las directivas adjuntadas a un grupo.
- Enumerar las políticas heredadas por un objeto (en virtud de las políticas adjuntadas a su grupo o uno de sus grupos principales).
- Configurar opciones de registro para objetos de un grupo. Consulte [Configuración de registros de AWS IoT \(p. 427\)](#).
- Crear trabajos que se enviarán y se ejecutarán en cada objeto del grupo y sus grupos secundarios. Consulte [Trabajos \(p. 675\)](#).

Los grupos de objetos estáticos tienen algunas limitaciones:

- Un grupo puede tener como máximo un grupo principal directo.
- Si un grupo es secundario de otro grupo, debe especificarlo en el momento en que se crea.
- No puede cambiar el principal de un grupo más adelante, así que asegúrese de planificar la jerarquía de grupos y crear un grupo principal antes de crear los grupos secundarios que contiene.
- El número de grupos a los que puede pertenecer un objeto es [limitado](#).
- No se puede agregar un objeto a más de un grupo en la misma jerarquía. (En otras palabras, no puede agregar un objeto a dos grupos que comparten un principal común).
- No se puede cambiar el nombre de un grupo.
- Los nombres de grupos de objetos no pueden contener caracteres internacionales, como û, é o ñ.
- No debe utilizar información personalmente identificable en el nombre de grupo de objetos. El nombre del grupo de cosas puede aparecer en comunicaciones e informes sin cifrar.
- No debe utilizar un carácter de dos puntos (:) en el nombre de un grupo de cosas. El carácter de dos puntos se utiliza como delimitador por otros AWS IoT y esto puede provocar que analicen cadenas con nombres de grupos de cosas incorrectamente.

Asociar políticas a grupos y separarlas de ellos puede mejorar la seguridad de las operaciones de AWS IoT de una serie de formas significativas. El método por dispositivo de asociar una política a un certificado, que luego se asocia a un objeto, consume mucho tiempo y dificulta la actualización o cambio rápido de políticas en una flota de dispositivos. Disponer de una política adjunta al grupo del objeto ahorra pasos cuando hay que rotar los certificados en un objeto. Además, las políticas se aplican dinámicamente a objetos cuando cambian la pertenencia a un grupo, por lo que no es necesario volver a crear un conjunto complejo de permisos cada vez que un dispositivo cambia la pertenencia en un grupo.

Crear un grupo de objetos estático

Utilice el comando `CreateThingGroup` para crear un grupo de objetos estático:

```
$ aws iot create-thing-group --thing-group-name LightBulbs
```

El comando `CreateThingGroup` devuelve una respuesta que contiene el nombre, el ID y el ARN del grupo de objetos estático:

```
{  
    "thingGroupName": "LightBulbs",  
    "thingGroupId": "abcdefghijklmnopqrstuvwxyz12345678ijklmnopqrstuvwxyz",  
    "thingGroupArn": "arn:aws:iot:us-west-2:123456789012:thinggroup/LightBulbs"  
}
```

Note

No es recomendable utilizar datos personales en los nombres de grupo de objetos.

A continuación, se muestra un ejemplo que especifica el grupo principal de un grupo de objetos estático durante su creación:

```
$ aws iot create-thing-group --thing-group-name RedLights --parent-group-name LightBulbs
```

Al igual que antes, el comando `CreateThingGroup` devuelve una respuesta que contiene el nombre, el ID y el ARN del grupo de objetos:

```
{  
    "thingGroupName": "RedLights",  
    "thingGroupId": "abcdefghijklmnopqrstuvwxyz12345678ijklmnopqrstuvwxyz",  
    "thingGroupArn": "arn:aws:iot:us-west-2:123456789012:thinggroup/RedLights"
```

```
        "thingGroupArn": "arn:aws:iot:us-west-2:123456789012:thinggroup/RedLights",  
    }
```

Important

Tenga en cuenta las siguientes limitaciones cuando cree jerarquías de grupos de objetos:

- Un grupo de objetos solo puede tener un grupo principal directo.
- El número de grupos secundarios directos que puede tener un grupo de objetos es [limitado](#).
- La profundidad máxima de una jerarquía de grupo es [limitada](#).
- El número de atributos que puede tener un grupo de objetos es [limitado](#). (Los atributos son pares nombre-valor que puede usar para almacenar información acerca de un grupo). La longitud de cada nombre de atributo y cada valor también es [limitada](#).

Descripción de un grupo de objetos

Puede utilizar el comando `DescribeThingGroup` para obtener información acerca de un grupo de objetos:

```
$ aws iot describe-thing-group --thing-group-name RedLights
```

El comando `DescribeThingGroup` devuelve información acerca del grupo especificado:

```
{  
    "thingGroupName": "RedLights",  
    "thingGroupArn": "arn:aws:iot:us-west-2:123456789012:thinggroup/RedLights",  
    "thingGroupId": "12345678abcdefghijklmnopqrstuvwxyz",  
    "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"  
    },  
}
```

Agregar un objeto a un grupo de objetos estático

Puede utilizar el comando `AddThingToThingGroup` para agregar un objeto a un grupo de objetos estático:

```
$ aws iot add-thing-to-thing-group --thing-name MyLightBulb --thing-group-name RedLights
```

El comando AddThingToThingGroup no genera una salida.

Important

Puede añadir un objeto a un máximo de 10 grupos. Sin embargo, no puede añadir un objeto a más de un grupo en la misma jerarquía. (En otras palabras, no puede agregar un objeto a dos grupos que comparten un principal común).

Si un objeto pertenece al número máximo de grupos de objetos posible y uno o varios de estos grupos es un grupo de objetos dinámico, puede utilizar la marca `overrideDynamicGroups` para que los grupos estáticos tengan prioridad sobre los grupos dinámicos.

Eliminar un objeto de un grupo de objetos estático

Puede utilizar el comando RemoveThingFromThingGroup para quitar un objeto de un grupo:

```
$ aws iot remove-thing-from-thing-group --thing-name MyLightBulb --thing-group-name RedLights
```

El comando RemoveThingFromThingGroup no genera una salida.

Enumerar los objetos en un grupo de objetos

Puede utilizar el comando ListThingsInThingGroup para listar los objetos que pertenecen a un grupo:

```
$ aws iot list-things-in-thing-group --thing-group-name LightBulbs
```

El comando ListThingsInThingGroup devuelve una lista de los objetos en el grupo determinado:

```
{
  "things": [
    "TestThingA"
  ]
}
```

Con el parámetro `--recursive`, puede enumerar los objetos que pertenecen a un grupo y también los que están en alguno de sus grupos secundarios:

```
$ aws iot list-things-in-thing-group --thing-group-name LightBulbs --recursive
```

```
{
  "things": [
    "TestThingA",
    "MyLightBulb"
  ]
}
```

Note

Esta operación es a [largo plazo coherente](#). En otras palabras, los cambios que se hagan en el grupo de objetos podrían no reflejarse inmediatamente.

Enumeración de grupos de objetos

Puede usar el comando ListThingGroups para mostrar los grupos de objetos de la cuenta:

```
$ aws iot list-thing-groups
```

La `ListThingGroups` devuelve una lista de grupos de objetos definidos en la cuenta de 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"
        }
    ]
}
```

Utilice los filtros opcionales para enumerar los grupos que tienen un grupo determinado como grupo principal (`--parent-group`) o los grupos cuyo nombre comienza por un prefijo determinado (`--name-prefix-filter`). El parámetro `--recursive` le permite listar también todos los grupos secundarios, no solo los grupos secundarios directos de un grupo de objetos:

```
$ aws iot list-thing-groups --parent-group LightBulbs
```

En este caso, el campo `ListThingGroups` devuelve una lista de grupos secundarios directos del grupo de objetos definido en la cuenta de AWS:

```
{
    "childGroups": [
        {
            "groupName": "RedLights",
            "groupArn": "arn:aws:iot:us-west-2:123456789012:thinggroup/RedLights"
        }
    ]
}
```

Utilice el parámetro `--recursive` con el comando `ListThingGroups` para listar todos los grupos secundarios de un grupo de objetos, no solo un elemento secundario directo:

```
$ aws iot list-thing-groups --parent-group LightBulbs --recursive
```

El comando `ListThingGroups` devuelve una lista de todos los grupos secundarios de un grupo de objetos:

```
{
    "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

Esta operación es a [largo plazo coherente](#). En otras palabras, los cambios que se hagan en el grupo de objetos podrían no reflejarse inmediatamente.

Enumerar grupos para un objeto

Puede utilizar el comando `ListThingGroupsForThing` para enumerar los grupos los que pertenece un objeto, incluidos los grupos principales:

```
$ aws iot list-thing-groups-for-thing --thing-name MyLightBulb
```

El comando `ListThingGroupsForThing` devuelve una lista de los grupos de objetos a los que pertenece este objeto, incluidos los grupos principales:

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

Actualizar un grupo de objetos estático

Puede utilizar el comando `UpdateThingGroup` para actualizar los atributos de un grupo de objetos estático:

```
$ aws iot update-thing-group --thing-group-name "LightBulbs" --thing-group-properties  
    "thingGroupDescription=\"this is a test group\", attributePayload=\"{\\"attributes  
\\\"=\\"Owner\\\"=\\"150\\\", \\"modelNames\\\"=\\"456\\\"}\\""
```

El comando `UpdateThingGroup` devuelve una respuesta que contiene el número de versión del grupo después de la actualización:

```
{  
    "version": 4  
}
```

Note

El número de atributos que un objeto puede tener es [limitado](#).

Eliminación de un grupo de objetos

Para eliminar un grupo de objetos, use el comando `DeleteThingGroup`:

```
$ aws iot delete-thing-group --thing-group-name "RedLights"
```

El comando `DeleteThingGroup` no genera una salida.

Important

Si intenta eliminar un grupo de objetos que tenga grupos secundarios de objetos, aparecerá un error:

```
A client error (InvalidRequestException) occurred when calling the  
DeleteThingGroup  
operation: Cannot delete thing group : RedLights when there are still child groups  
attached to it.
```

Debe eliminar los grupos secundarios antes de eliminar el grupo.

Puede eliminar un grupo que tenga objetos secundarios, pero no se seguirán aplicando los permisos concedidos a los objetos por la pertenencia del grupo. Antes de eliminar un grupo que tenga una política adjunta, compruebe detenidamente que la eliminación de esos permisos no hará que los objetos del grupo no funcionen correctamente. Además, tenga en cuenta que los comandos que muestran a qué grupos pertenece un objeto (por ejemplo, `ListGroupsForThing`) podrían seguir mostrando el grupo mientras se actualizan los registros en la nube.

Asociar una política a un grupo de objetos estático

Puede usar el comando `AttachPolicy` para asociar una política a un grupo de objetos estático y, por extensión, a todos los objetos de ese grupo y de sus grupos secundarios:

```
$ aws iot attach-policy \  
--target "arn:aws:iot:us-west-2:123456789012:thinggroup/LightBulbs" \  
--policy-name "myLightBulbPolicy"
```

El comando `AttachPolicy` no genera una salida

Important

Puede asociar un número máximo de dos políticas a un grupo.

Note

No es recomendable utilizar datos personales en los nombres de política.

La `--target` puede ser un ARN de grupo de objetos (como más arriba), un ARN del certificado o una identidad de Amazon Cognito. Para obtener más información acerca de las políticas, los certificados y la autenticación, consulte [Autenticación \(p. 294\)](#).

Para obtener más información, consulte [AWS IoT Core Políticas de](#).

Desconectar una política de un grupo de objetos estático

Puede usar el comando `DetachPolicy` para separar una política de un grupo de objetos y de esa forma, por extensión, a todos los objetos de ese grupo y a los objetos de cualquiera de sus grupos secundarios:

```
$ aws iot detach-policy --target "arn:aws:iot:us-west-2:123456789012:thinggroup/LightBulbs"  
--policy-name "myLightBulbPolicy"
```

El comando `DetachPolicy` no genera una salida.

Mostrar las políticas asociadas a un grupo de objetos estático

Puede utilizar el comando `ListAttachedPolicies` para mostrar las políticas asociadas a un grupo de objetos estático:

```
$ aws iot list-attached-policies --target "arn:aws:iot:us-west-2:123456789012:thinggroup/  
RedLights"
```

La `--target` puede ser un ARN de grupo de objetos (como más arriba), un ARN del certificado o una identidad de Amazon Cognito.

Agregue el parámetro `--recursive` opcional para incluir también todas las políticas asociadas a los grupos principales del grupo.

El comando `ListAttachedPolicies` devuelve una lista de políticas:

```
{  
    "policies": [  
        "MyLightBulbPolicy"  
        ...  
    ]  
}
```

Enumeración de los grupos para una política

Puede utilizar el comando `ListTargetsForPolicy` para enumerar los destinos, incluidos los grupos, a los que se adjunta una política:

```
$ aws iot list-targets-for-policy --policy-name "MyLightBulbPolicy"
```

Añada el parámetro `--page-size number` opcional para especificar el número máximo de resultados que se va a devolver para cada consulta y el parámetro `--marker string` en las llamadas siguientes para recuperar el siguiente conjunto de resultados, si lo hubiera.

El comando `ListTargetsForPolicy` devuelve una lista de destinos y el token que usar para recuperar más resultados:

```
{  
    "nextMarker": "string",  
    "targets": [ "string" ... ]
```

}

Obtención de políticas en vigor para un objeto

Puede usar el comando GetEffectivePolicies para mostrar las políticas en vigor de un objeto, incluidas las políticas asociadas a grupos a los que pertenece el objeto (si el grupo es un grupo principal directo o un antecesor indirecto):

```
$ aws iot get-effective-policies \
--thing-name "MyLightBulb" \
--principal "arn:aws:iot:us-east-1:123456789012:cert/
a0c01f5835079de0a7514643d68ef8414ab739a1e94ee4162977b02b12842847"
```

Utilice el parámetro --principal para especificar el ARN del certificado adjunto al objeto. Si utiliza la autenticación de identidad de Amazon Cognito, utilice el --cognito-identity-pool-id, opcionalmente, agregue el --principal para especificar una identidad de Amazon Cognito. Si especifica solo el --cognito-identity-pool-id, se devuelven las políticas asociadas a ese rol del grupo de identidades para usuarios sin autenticar. Si usa ambos, se devuelven las políticas asociadas a ese rol del grupo de identidades para los usuarios autenticados.

El parámetro --thing-name es opcional y puede usarse en lugar del parámetro --principal. Cuando se utiliza, se devolverán las políticas asociadas a cualquier grupo al que pertenece el objeto y las políticas asociadas a grupos principales de estos grupos (hasta el grupo raíz en la jerarquía).

El comando GetEffectivePolicies devuelve una lista de políticas:

```
{
  "effectivePolicies": [
    {
      "policyArn": "string",
      "policyDocument": "string",
      "policyName": "string"
    }
    ...
  ]
}
```

Prueba de autorización para acciones de MQTT

Puede utilizar el TestAuthorization para comprobar si una acción (Publish,Subscribe) está permitido para algo:

```
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\"]}"
```

Utilice el parámetro --principal para especificar el ARN del certificado adjunto al objeto. Si usa la autenticación de Amazon Cognito, especifique una identidad de Cognito como --principal o utilice el --cognito-identity-pool-id parámetro, o ambos. (Si especifica solo --cognito-identity-pool-id, se tienen en cuenta las políticas asociadas a ese rol del grupo de identidades para usuarios sin autenticar. Si usa ambos, se tienen en cuenta las políticas asociadas a ese rol del grupo de identidades para los usuarios autenticados).

Especifique una o más acciones de MQTT que desee probar mediante la enumeración de conjuntos de recursos y tipos de acción que siguen al parámetro --auth-infos. El campo actionType debería

contener "PUBLISH", "SUBSCRIBE", "RECEIVE" o "CONNECT". El campo `resources` debe contener una lista de ARN de recursos. Para obtener más información, consulte [Políticas de AWS IoT Core \(p. 333\)](#).

Puede probar los efectos de la adición de políticas mediante la especificación de estas con el parámetro `--policy-names-to-add`. También puede probar los efectos de la eliminación de políticas mediante ellas con el parámetro `--policy-names-to-skip`.

Puede usar el parámetro `--client-id` opcional para restringir los resultados.

El comando `TestAuthorization` devuelve detalles acerca de acciones que se permiten o deniegan para cada conjunto de consultas `--auth-infos` especificadas:

```
{  
    "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 objetos dinámicos

Note

La función de indexación de flotas para admitir la indexación denominada sombras y AWS IoT Device DefenderLos datos de infracciones están en versión preliminar paraAWS IoTAdministración de dispositivos y está sujeta a cambios.

Los grupos de objetos dinámicos actualizan la pertenencia al grupo a través de consultas de búsqueda. Los grupos de objetos dinámicos permiten cambiar la forma en que se interactúa con los objetos en función de los datos de vulneración de conectividad, registro, sombra o Device Defender. Puesto que los grupos de objetos dinámicos están vinculados al índice de su flota, debe habilitar la indexación de flotas para utilizarlos. Puede obtener una vista previa de los objetos en un grupo de objetos dinámico antes de crear el grupo con una consulta de búsqueda de indexación de flotas. Para obtener más información, consulte [Indexación de flotas \(p. 825\)](#) y [Sintaxis de la consulta \(p. 846\)](#).

Puede especificar un grupo de objetos dinámico como destino para un trabajo. Solo realizan el trabajo los objetos que cumplen los criterios que definen el grupo de objetos dinámico.

Por ejemplo, supongamos que desea actualizar el firmware en sus dispositivos, pero, para reducir la posibilidad de que la actualización se interrumpa, solo desea actualizar el firmware en los dispositivos con duración de la batería superior al 80 %. Puede crear un grupo de objetos dinámico que solo incluya dispositivos con una duración de la batería por encima del 80 %, y puede utilizar el grupo de objetos dinámico como destino para el trabajo de actualización del firmware. Recibirán la actualización del firmware solo los dispositivos que cumplan con sus criterios de duración de la batería. A medida que los dispositivos alcancen el 80 % de duración de la batería, estos se van añadiendo al grupo de objetos dinámico y reciben la actualización del firmware.

Para obtener más información sobre cómo especificar grupos de objetos como destinos de trabajos, consulte [CreateJob](#).

Los grupos de objetos dinámicos se diferencian de los grupos de objetos estáticos como sigue:

- La pertenencia de un objeto no se define de forma explícita. Para crear un grupo de objetos dinámico, debe definir [una cadena de consulta \(p. 847\)](#) que define la pertenencia a los grupos.
- Los grupos de objetos dinámicos no pueden formar parte de una jerarquía.
- Los grupos de objetos dinámicos no pueden tener políticas aplicadas.
- Puede utilizar un conjunto diferente de comandos para crear, actualizar y eliminar grupos de objetos dinámicos. Para el resto de operaciones, para interactuar con grupos de objetos dinámicos se pueden usar los mismos comandos que se utilizan para interactuar con grupos de objetos estáticos.
- El número de grupos dinámicos que puede tener una sola cuenta es [limitado](#).
- No debe utilizar información personalmente identificable en el nombre de grupo de objetos. El nombre del grupo de cosas puede aparecer en comunicaciones e informes sin cifrar.
- No debe utilizar un carácter de dos puntos (:) en el nombre de un grupo de cosas. El carácter de dos puntos se utiliza como delimitador por otros AWS IoT; esto puede provocar que analicen cadenas con nombres de grupos de cosas incorrectamente.

Para obtener más información acerca de los grupos de objetos estáticos, consulte [Grupos de objetos estáticos \(p. 273\)](#).

Por ejemplo, suponga que creamos un grupo dinámico que contiene todas las salas de un almacén cuya temperatura es superior a 60 grados Fahrenheit. Cuando la temperatura de una sala sea 61 grados o superior, esta sala se añade al grupo de objetos dinámico RoomTooWarm. Todas las salas del grupo de objetos dinámico RoomTooWarm tienen encendidos los ventiladores. Cuando la temperatura de una sala cae a 60 grados o menos, se elimina del grupo de objetos dinámico y se apaga su ventilador.

Crear un grupo de objetos dinámico

Utilice el comando `CreateDynamicThingGroup` para crear un grupo de objetos dinámico. Para crear un grupo de objetos dinámico para el escenario de la sala demasiado caliente, utilizaría el comando `create-dynamic-thing-group` de la CLI:

```
$ aws iot create-dynamic-thing-group --thing-group-name "RoomTooWarm" --query-string  
"attributes.temperature>60"
```

Note

No es recomendable utilizar datos personales en los nombres de grupo de objetos dinámico.

La `CreateDynamicThingGroup` devuelve una respuesta que contiene el nombre del índice, la cadena de consulta, la versión de la consulta, el nombre del grupo de objetos, el ID del grupo de objetos y el nombre de recurso de Amazon (ARN) de su grupo de objetos:

```
{  
    "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": "abcdefg12345678ijklmnop12345678qrstuvwxyz"  
}
```

El grupo de objetos dinámico no se crea de manera instantánea. Reponer un grupo de objetos dinámico lleva tiempo. Cuando se crea un grupo de objetos dinámico, el estado del grupo se establece en `BUILDING`. Cuando termina el proceso de reposición, el estado cambia a `ACTIVE`. Para comprobar el estado del grupo de objetos dinámico, utilice el comando [DescribeThingGroup](#).

Describir un grupo de objetos dinámico

Utilice el comando `DescribeThingGroup` para obtener información acerca de un grupo de objetos dinámico:

```
$ aws iot describe-thing-group --thing-group-name "RoomTooWarm"
```

El comando `DescribeThingGroup` devuelve información acerca del 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"  
}
```

Al ejecutar `DescribeThingGroup` en un grupo de objetos dinámico se devuelven atributos específicos de grupos de objetos dinámicos, como, por ejemplo, `queryString` y el estado.

El estado de un grupo de objetos dinámico puede tomar los siguientes valores:

`ACTIVE`

El grupo de objetos dinámico está listo para usarse.

BUILDING

Se está creando el grupo de objetos dinámico y se está procesando la pertenencia de los objetos.

REBUILDING

Se está actualizando la pertenencia al grupo de objetos dinámico tras ajustar la consulta de búsqueda del grupo.

Note

Tras crear un grupo de objetos dinámico podrá utilizar el grupo, independientemente de su estado. Solo los grupos de objetos dinámicos con estado ACTIVE incluyen todos los objetos que coinciden con la consulta de búsqueda para ese grupo de objetos dinámico. Los grupos de objetos dinámicos con estado BUILDING y REBUILDING podrían no incluir todos los objetos coincidentes con la consulta de búsqueda.

Actualizar un grupo de objetos dinámico

Utilice el comando `UpdateDynamicThingGroup` para actualizar los atributos de un grupo de objetos dinámico, incluida la consulta de búsqueda del grupo. El siguiente comando actualiza la descripción del grupo de objetos y la cadena de consulta que cambia los criterios de pertenencia a una 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"
```

El comando `UpdateDynamicThingGroup` devuelve una respuesta que contiene el número de versión del grupo después de la actualización:

```
{  
    "version": 2  
}
```

Los grupos de objetos dinámicos no se actualizan de manera instantánea. Reponer un grupo de objetos dinámico lleva tiempo. Cuando un grupo de objetos dinámico se actualiza, su estado cambia a REBUILDING, al tiempo que el grupo actualiza su pertenencia. Cuando termina el proceso de reposición, el estado cambia a ACTIVE. Para comprobar el estado del grupo de objetos dinámico, utilice el comando [DescribeThingGroup](#).

Eliminar un grupo de objetos dinámico

Para eliminar un grupo de objetos dinámico, use el comando `DeleteDynamicThingGroup`:

```
$ aws iot delete-dynamic-thing-group --thing-group-name "RoomTooWarm"
```

El comando `DeleteDynamicThingGroup` no genera una salida.

Los comandos que muestran a qué grupos pertenece un objeto (por ejemplo, `ListGroupsForThing`) podrían seguir mostrando el grupo mientras se actualizan los registros en la nube.

Limitaciones y conflictos

Los grupos de objetos dinámicos comparten estas limitaciones con los grupos de objetos estáticos:

- El número de atributos que puede tener un grupo de objetos es [limitado](#).
- El número de grupos a los que puede pertenecer un objeto es [limitado](#).
- A los grupos de objetos no se les puede cambiar el nombre.
- Los nombres de grupos de objetos no pueden contener caracteres internacionales, como û, é o ñ.

A la hora de utilizar grupos de objetos dinámicos, tenga en cuenta lo siguiente:

El servicio de indexación de flotas debe estar habilitado.

El servicio de indexación de flotas debe estar habilitado y el proceso de reposición de indexación de flotas debe haberse completado para poder crear y utilizar grupos de objetos dinámicos. Normalmente se producirá cierto retraso tras habilitar el servicio de indexación de flotas. El proceso de reposición puede llevar algún tiempo. Cuantos más objetos haya registrado, más tiempo tardará en completarse el proceso de reposición. Tras habilitar el servicio de indexación de flotas para grupos de objetos dinámicos, no podrá deshabilitarlo hasta que se eliminen todos los grupos de objetos dinámicos.

Note

Si tiene los permisos para consultar el índice de la flota, podrá obtener acceso a los datos de los objetos en toda la flota.

El número de grupos de objetos dinámicos es limitado

El número de grupos dinámicos es [limitado](#).

Comandos correctos pueden registrar errores

Cuando se crea o actualiza un grupo de objetos dinámico, es posible que algunos de los objetos válidos para estar en el grupo dinámico no se hayan agregado todavía. Sin embargo, el comando para crear o actualizar un grupo de cosas dinámico sigue ejecutándose correctamente en esos casos, aunque registran un error y generan una [métrica AddThingToDynamicThingGroupsFailed \(p. 438\)](#).

Un [entrada de registro de errores](#) en el registro de CloudWatch se crea para cada cosa cuando no se puede agregar una cosa apta a un grupo de cosas dinámico o se elimina una cosa de un grupo de cosas dinámico para agregarlo a otro grupo. Cuando no se puede añadir algo a un grupo dinámico, un [AddThingToDynamicThingGroupsFailed métrica \(p. 438\)](#) también se crea; sin embargo, una sola métrica puede representar varias entradas de registro.

Cuando un objeto pasa a ser válido para agregarse a un grupo de objetos dinámico, se tiene en cuenta lo siguiente:

- ¿Este objeto ya está en el número máximo de grupos posible? (Consulte los [límites](#))
 - NO: El objeto se agrega al grupo de objetos dinámico.
 - Sí: ¿El objeto es miembro de algún grupo de objetos dinámico?
 - NO: El objeto no se puede agregar al grupo de objetos dinámico, se registra un error y un [AddThingToDynamicThingGroupsFailed métrica \(p. 438\)](#) se genera.
 - Sí: ¿El grupo de objetos dinámico que se va a unir es anterior a cualquier otro grupo de objetos dinámicos del que el objeto ya sea miembro?
 - NO: El objeto no se puede agregar al grupo de objetos dinámico, se registra un error y un [AddThingToDynamicThingGroupsFailed métrica \(p. 438\)](#) se genera.
 - Sí: Elimine el objeto del grupo de objetos dinámicos más reciente del que es miembro, registre un error y agregue el objeto al grupo de objetos dinámicos. Esto genera un error y una [métrica AddThingToDynamicThingGroupsFailed \(p. 438\)](#) en el grupo de objetos dinámicos del que se ha eliminado el objeto.

Cuando un objeto de un grupo dinámico ya no satisface la consulta de búsqueda, se elimina del grupo de objetos dinámicos. Del mismo modo, cuando un objeto se actualiza para satisfacer una consulta de búsqueda de un grupo de objetos dinámico, se agrega al grupo tal y como se describió anteriormente. Estas incorporaciones y eliminaciones son normales y no generan entradas en el registro de errores.

Con `overrideDynamicGroups` habilitado, los grupos estáticos disfrutan de prioridad sobre los grupos dinámicos.

El número de grupos a los que puede pertenecer un objeto es [limitado](#). Cuando se actualiza la pertenencia de un objeto utilizando el comando `AddThingToThingGroup` o `UpdateThingGroupsForThing`, si se agrega el parámetro `--overrideDynamicGroups`, se otorga prioridad a los grupos de objetos estáticos frente a los dinámicos.

Cuando se agrega un objeto a un grupo de objetos estático, debe tenerse en cuenta lo siguiente:

- ¿El objeto ya es miembro del número máximo de grupos?
 - NO: El objeto se agrega al grupo de objetos estático.
 - SÍ: ¿El objeto está en algún grupo dinámico?
 - NO: El objeto no se puede agregar al grupo de objetos. El comando genera una excepción.
 - SÍ: Era--overrideDynamicGroupsestá habilitado?
 - NO: El objeto no se puede agregar al grupo de objetos. El comando genera una excepción.
 - SÍ: El objeto se elimina del último grupo de objetos dinámicos que se creó, se registra un error y un[AddThingToDynamicThingGroupsFailed](#)métrica (p. 438)se genera para el grupo de objetos dinámico del que se ha eliminado el objeto. A continuación, el objeto se agrega al grupo de objetos estáticos.

Los grupos de objetos dinámicos más antiguos tienen prioridad sobre los más nuevos.

El número de grupos a los que puede pertenecer un objeto es [limitado](#). Cuando un objeto pasa a ser válido para agregarse a un grupo de objetos dinámico gracias a una operación de creación o actualización pero ya está en el máximo número de grupos posible, puede eliminarse de otro grupo de objetos dinámico y agregarse en este. Si necesita más información sobre esto, consulte [Comandos correctos pueden registrar errores \(p. 286\)](#) y [Con `overrideDynamicGroups` habilitado, los grupos estáticos disfrutan de prioridad sobre los grupos dinámicos. \(p. 287\)](#) para ver ejemplos.

Cuando un objeto se elimina de un grupo de objetos dinámico, se registra un error y se genera un evento.

No se pueden aplicar políticas a grupos de objetos dinámicos

Si intenta aplicar una política a un grupo de objetos dinámico, se generará una excepción.

La pertenencia a grupos de objetos dinámicos es coherente a largo plazo.

Para el registro solo se evalúa el estado final de un objeto. Los estados intermedios se pueden omitir si se actualizan rápidamente los estados. Evite asociar una regla, un trabajo con un grupo de objetos dinámico cuya pertenencia dependa de un estado intermedio.

Etiquetado de los recursos de AWS IoT

Para administrar y organizar más fácilmente grupos de objetos, tipos de objetos, reglas de temas, trabajos, auditorías programadas y perfiles de seguridad, dispone de la opción de asignar sus propios metadatos en forma de etiquetas a cada uno de estos recursos. En esta sección se describe qué son las etiquetas y cómo crearlas.

Para ayudarle a administrar los costos relacionados con los objetos puede crear [grupos de facturación \(p. 291\)](#) con objetos. A continuación, puede asignar etiquetas con sus metadatos a cada uno de estos grupos de facturación. En esta sección se explican también los grupos de facturación y los comandos disponibles para crearlos y administrarlos.

Conceptos básicos de etiquetas

Puede utilizar etiquetas para clasificar los recursos de AWS IoT de diversas maneras (por ejemplo, según su finalidad, su propietario o su entorno). Esto resulta útil cuando tiene muchos recursos del mismo tipo, ya que le permite identificar rápidamente un recurso específico utilizando las etiquetas asignadas. Cada etiqueta está formada por una clave y un valor opcional, ambos definidos por el usuario. Por ejemplo, puede definir un conjunto de etiquetas para los tipos de objetos de modo que le resulte más fácil hacer un seguimiento de los dispositivos con arreglo al tipo al que pertenecen. Le recomendamos que cree un conjunto de claves de etiqueta que cumpla sus necesidades para cada tipo de recurso. Mediante el uso de un conjunto coherente de claves de etiquetas, podrá administrar los recursos más fácilmente.

Puede buscar y filtrar sus recursos en función de las etiquetas que añada o aplique. También puede utilizar etiquetas de grupo de facturación para categorizar y realizar un seguimiento de los costos. También puede utilizar etiquetas para controlar el acceso a los recursos según se describe en [Uso de etiquetas con políticas de IAM \(p. 289\)](#).

Para facilitar el uso, el Editor de etiquetas en la AWS Management Console proporciona una forma unificada y centralizada de crear y administrar sus etiquetas. Para obtener más información, consulte [Uso de Tag Editor en la AWS Consola de administración de](#).

También puede trabajar con etiquetas utilizando la AWS CLI y la AWS IoT API. Puede asociar etiquetas a grupos de objetos, tipos de objetos, reglas de temas, trabajos, perfiles de seguridad, políticas y grupos de facturación en el momento de crearlos utilizando la `Tags` en los siguientes comandos:

- [CreateBillingGroup](#)
- [Crear destino](#)
- [Crear perfil de dispositivo](#)
- [CreateDynamicThingGroup](#)
- [CreateJob](#)
- [CreateOTAUpdate](#)
- [CreatePolicy](#)
- [CreateScheduledAudit](#)
- [CreateSecurityProfile](#)

- [Crear perfil de servicio](#)
- [CreateStream](#)
- [CreateThingGroup](#)
- [CreateThingType](#)
- [CreateTopicRule](#)
- [Crear puerta de enlace inalámbrica](#)
- [Crear dispositivo inalámbrico](#)

Puede añadir, modificar o eliminar etiquetas para recursos existentes que admitan el uso de etiquetas con los siguientes comandos:

- [TagResource](#)
- [ListTagsForResource](#)
- [UntagResource](#)

Puede editar las claves y los valores de las etiquetas y también puede eliminar etiquetas de un recurso en cualquier momento. Puede establecer el valor de una etiqueta como una cadena vacía, pero no puede asignarle un valor nulo. Si añade una etiqueta con la misma clave que una etiqueta existente en ese recurso, el nuevo valor sobrescribirá al antiguo. Si elimina un recurso, también se eliminarán todas las etiquetas que este tenga asociadas.

Restricciones y limitaciones en las etiquetas

Se aplican las siguientes restricciones básicas a las etiquetas:

- Número máximo de etiquetas por recurso: 50
- Longitud máxima de la clave: 127 caracteres Unicode en UTF-8
- Longitud máxima del valor: 255 caracteres Unicode en UTF-8
- Las claves y los valores de las etiquetas distinguen entre mayúsculas y minúsculas.
- No utilizar el prefijo `:aws` en los nombres o valores de etiqueta, ya que su uso está reservado a AWS. Los nombres y valores de etiquetas que tienen este prefijo no se pueden editar. Las etiquetas que tengan este prefijo no cuentan para el límite de etiquetas por recurso.
- Si se utiliza su esquema de etiquetado en múltiples servicios y recursos, recuerde que otros servicios podrían tener otras restricciones sobre caracteres permitidos. Los caracteres permitidos son: letras, espacios y números representables en UTF-8, además de los siguientes caracteres especiales: + - = . _ : / @.

Uso de etiquetas con políticas de IAM

Puede aplicar permisos de nivel de recurso basados en etiquetas en las políticas de IAM que utiliza con las acciones de la API de AWS IoT. Esto le ofrece un mejor control sobre los recursos que un usuario puede crear, modificar o utilizar. Puede utilizar el elemento `Condition` (también llamado bloque `Condition`) junto con las siguientes claves de contexto de condición y valores en una política de IAM para controlar el acceso del usuario (permiso) en función de las etiquetas de un usuario:

- Utilice `aws:ResourceTag/tag-key: tag-value` para permitir o denegar acciones de los usuarios en recursos con etiquetas específicas.
- Utilice `aws:RequestTag/tag-key: tag-value` para exigir (o impedir) el uso de una etiqueta específica al realizar una solicitud de API para crear o modificar un recurso que permita etiquetas.

- Utilice `aws:TagKeys: [tag-key, ...]` para exigir (o impedir) el uso de un conjunto de claves de etiquetas al realizar una solicitud de API para crear o modificar un recurso que permita etiquetas.

Note

Las claves de contexto de condición y los valores de una política de IAM se aplican únicamente a las acciones de AWS IoT en las que un identificador de un recurso que se puede etiquetar es un parámetro obligatorio. Por ejemplo, no se puede permitir ni denegar el uso de `DescribeEndpoint` en función de los valores y las claves de contexto de condición porque en esta solicitud no se hace referencia a ningún recurso que se pueda etiquetar (grupos de objetos, tipos de objetos, reglas de temas, trabajos o perfiles de seguridad). Para obtener más información acerca de [AWS IoT Recursos que se pueden etiquetar y las claves de condición que admiten](#), leer [Acciones, recursos y claves de condición para AWS IoT](#).

Para obtener más información acerca del uso de etiquetas, consulte [Control del acceso mediante etiquetas](#) en la AWS Identity and Access Management Guía del usuario de. La sección de [referencia de políticas JSON de IAM](#) de esta guía incluye sintaxis, descripciones y ejemplos detallados de los elementos, variables y lógica de evaluación de las políticas JSON de IAM.

La siguiente política de ejemplo aplica dos restricciones basadas en etiquetas para la `ThingGroup`. Un usuario de IAM restringido por esta política:

- No se puede crear un grupo de objetos la etiqueta «env=prod» (en el ejemplo, consulte la línea). `"aws:RequestTag/env" : "prod"`.
- No se puede modificar u obtener acceso a un grupo de objetos que tiene una etiqueta existente «env=prod» (en el ejemplo, consulte la línea). `"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": "*"
}
}
```

También puede especificar varios valores de etiqueta para una determinada clave de etiqueta encerrándola en una lista, tal y como se muestra a continuación:

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

Note

Si permite o deniega a los usuarios acceso a recursos en función de etiquetas, debe considerar denegar explícitamente a los usuarios la posibilidad de agregar estas etiquetas o retirarlas de los mismos recursos. De lo contrario, es posible que un usuario eluda sus restricciones y obtenga acceso a un recurso modificando sus etiquetas.

Grupos de facturación

AWS IoT no le permite aplicar directamente etiquetas a objetos individuales, pero sí le permite colocar objetos en grupos de facturación y aplicarles etiquetas. En AWS IoT, la asignación de datos de costos y uso basados en etiquetas queda limitada a los grupos de facturación.

AWS IoT Corepara los recursos de LoraWAN, como dispositivos inalámbricos y puertas de enlace, no se pueden agregar a los grupos de facturación. Sin embargo, se pueden asociar conAWS IoTcosas, que se pueden agregar a los grupos de facturación.

Dispone de los siguientes comandos:

- [AddThingToBillingGroup](#) agrega un objeto a un grupo de facturación.
- [CreateBillingGroup](#) crea un grupo de facturación.
- [DeleteBillingGroup](#) elimina el grupo de facturación.
- [DescribeBillingGroup](#) devuelve información sobre un grupo de facturación.
- [ListBillingGroups](#) muestra los grupos de facturación que ha creado.
- [ListThingsInBillingGroup](#) muestra los objetos que ha agregado al grupo de facturación indicado.
- [RemoveThingFromBillingGroup](#) elimina el objeto indicado del grupo de facturación.
- [UpdateBillingGroup](#) actualiza la información sobre el grupo de facturación.
- [CreateThing](#) permite especificar un grupo de facturación para el objeto durante su creación.
- [DescribeThing](#) devuelve la descripción de un objeto, incluido el grupo de facturación al que este pertenece, si lo hay.

LaAWS IoTLa API inalámbrica proporciona estas acciones para asociar dispositivos inalámbricos y puertas de enlace conAWS IoTobjetos.

- [Asociar dispositivo inalámbrico con cosa](#)
- [Asociar puerta de enlace inalámbrica con cosa](#)

Visualización de datos de uso y asignación de costos

Puede utilizar etiquetas de grupo de facturación para categorizar y realizar un seguimiento de los costos. Cuando aplica etiquetas a grupos de facturación (y, por lo tanto, a las cosas que incluyen), AWS genera un informe de asignación de costos como un archivo CSV (valores separados por comas) con el total del uso y los costos por etiqueta. Puede aplicar etiquetas que representen categorías de negocio (por ejemplo, centros de costos, nombres de aplicación o propietarios) para organizar los costos entre diferentes servicios. Para obtener más información acerca de utilizar etiquetas para asignación de costos, consulte [Uso de etiquetas de asignación de costos](#) en la [AWS Guía del usuario Billing and Cost Management de costos](#).

Note

Para asociar los datos de uso y costos de forma precisa con los objetos que ha colocado en grupos de facturación, cada dispositivo o cada aplicación debe:

- Haberse registrado como objeto en AWS IoT. Para obtener más información, consulte [Administración de dispositivos con AWS IoT \(p. 266\)](#).
- Conéctese al agente de mensajes de AWS IoT a través de MQTT utilizando únicamente el nombre del objeto como ID de cliente. Para obtener más información, consulte [the section called “Protocolos de comunicación de dispositivos” \(p. 81\)](#).
- Autenticarse mediante un certificado de cliente asociado al objeto.

Dispone de las dimensiones de precios para los grupos de facturación (en función de la actividad de los objetos asociados al grupo de facturación):

- Conectividad (en función del nombre del objeto utilizado como ID de cliente para conectarse)
- Mensajes (en función de la entrada de mensajes procedentes de un objeto y la salida de mensajes destinados a un objeto; solo MQTT)
- Operaciones de sombra (en función del objeto cuyo mensaje activó una actualización de sombra)
- Reglas activadas (en función del objeto cuyo mensaje entrante activó la regla; no se aplica a las reglas activadas por los eventos del ciclo de vida de MQTT)
- Actualizaciones de índices de objetos (en función del objeto que se haya agregado al índice)
- Acciones remotas (en función del objeto actualizado)
- Informes de [Detect \(p. 983\)](#) (en función del objeto cuya actividad se notifica).

Los datos de costo y uso basados en etiquetas (y notificados para un grupo de facturación) no reflejan las siguientes actividades:

- Operaciones de registro de dispositivos (incluidas las actualizaciones de objetos, grupos de objetos y tipos de objetos). Para obtener más información, consulte [Administración de dispositivos con AWS IoT \(p. 266\)](#).
- Actualizaciones de índices de grupos de objetos (al agregar un grupo de objetos)
- Consultas de búsqueda en índices.
- [Aprovisionamiento de dispositivos \(p. 795\)](#).
- Informes de [Auditoría \(p. 919\)](#).

Seguridad en AWS IoT

La seguridad en la nube de AWS es la mayor prioridad. Como cliente de AWS, se beneficia de una arquitectura de red y un centro de datos que se han diseñado para satisfacer los requisitos de seguridad de las organizaciones más exigentes.

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

- Seguridad de la nube: AWS es responsable de proteger la infraestructura que ejecuta los servicios de AWS en la nube de AWS. AWS también proporciona servicios que puede utilizar de forma segura. Auditores externos prueban y verifican periódicamente la eficacia de nuestra seguridad en el marco de los [programas de conformidad de AWS](#). Para obtener más información sobre los programas de conformidad que se aplican a AWS IoT, consulte [Servicios de AWS en el ámbito del programa de conformidad](#).
- Seguridad en la nube: su responsabilidad viene determinada por el servicio de AWS que utilice. También es responsable de otros factores, incluida la confidencialidad de los datos, los requisitos de la empresa y la legislación y los reglamentos vigentes.

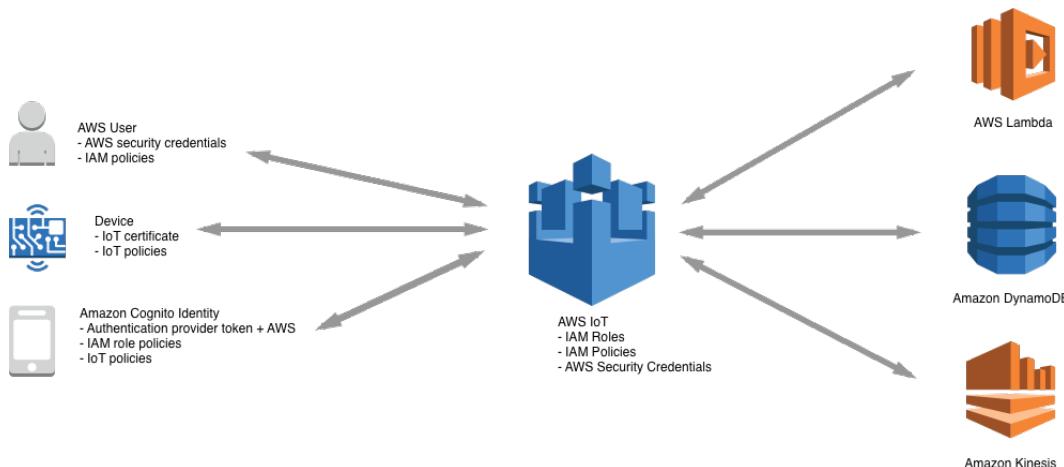
Esta documentación le ayuda a comprender cómo aplicar el modelo de responsabilidad compartida cuando se utiliza AWS IoT. En los siguientes temas, se le mostrará cómo configurar AWS IoT para satisfacer sus objetivos de seguridad y conformidad. También puede aprender a utilizar otros servicios de AWS que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

Temas

- [Seguridad de AWS IoT \(p. 293\)](#)
- [Autenticación \(p. 294\)](#)
- [Autorización \(p. 331\)](#)
- [Protección de los datos en AWS IoT Core \(p. 381\)](#)
- [Identity and Access Management en AWS IoT \(p. 384\)](#)
- [Registro y monitorización \(p. 413\)](#)
- [Validación de la conformidad enAWS IoT Núcleo \(p. 415\)](#)
- [Resiliencia enAWS Núcleo de IoT \(p. 415\)](#)
- [Uso de AWS IoT Core con los puntos de enlace de la VPC de la interfaz \(p. 416\)](#)
- [Seguridad de la infraestructura en AWS IoT \(p. 419\)](#)
- [Supervisión de seguridad de flotas o dispositivos de producción conAWS IoT Núcleo \(p. 419\)](#)
- [Prácticas de seguridad recomendadas para AWS IoT Core \(p. 419\)](#)
- [AWS Training and Certification \(p. 425\)](#)

Seguridad de AWS IoT

Cada dispositivo o cliente conectado debe tener una credencial con la que interactuar con AWS IoT. Todo el tráfico hacia y desde AWS IoT se envía de forma segura mediante Transport Layer Security (TLS). Los mecanismos de seguridad de la nube protegen los datos cuando estos avanzan AWS IoT y otros servicios de AWS.



- Usted es el responsable de administrar las credenciales de dispositivos (certificados X.509, AWS credenciales, identidades de Amazon Cognito, identidades federadas o tokens de autenticación personalizados) y políticas en AWS IoT. Para obtener más información, consulte [Administración de claves en AWS IoT \(p. 384\)](#). También es el responsable de asignar identidades exclusivas a cada uno de los dispositivos y de administrar los permisos de cada dispositivo o un grupo de dispositivos.
- Los dispositivos se conectan a AWS IoT utilizando certificados X.509 o identidades de Amazon Cognito en una conexión TLS segura. Durante la investigación y el desarrollo, así como para algunas aplicaciones que realizan llamadas a la API o utilizan WebSockets también puede autenticar utilizando usuarios y grupos de IAM o tokens de autenticación personalizados. Para obtener más información, consulte [Usuarios, grupos y roles de IAM \(p. 318\)](#).
- Al utilizar la autenticación de AWS IoT, el agente de mensajes es responsable de autenticar los dispositivos, de incorporar los datos del dispositivo de forma segura y de conceder o denegar los permisos de acceso que especifique para sus dispositivos mediante políticas de AWS IoT.
- Cuando se utiliza la autenticación personalizada, un autorizador personalizado es responsable de autenticar los dispositivos y de conceder o denegar los permisos de acceso que especifique para los dispositivos mediante AWS IoT políticas de IAM.
- La AWS IoT Rules engine reenvía los datos de los dispositivos a otros dispositivos u otros AWS servicios según las reglas que defina. Utiliza AWS Identity and Access Management para transferir datos de forma segura a su destino final. Para obtener más información, consulte [Identity and Access Management en AWS IoT \(p. 384\)](#).

Autenticación

La autenticación es un mecanismo en el que se verifica la identidad de un cliente o un servidor. La autenticación del servidor es el proceso en el que los dispositivos u otros clientes aseguran que se comunican con un punto de enlace de AWS IoT real. La autenticación de cliente es el proceso en el que los dispositivos u otros clientes se autentican con AWS IoT.

AWS Training and Certification

Siga el siguiente curso para obtener más información sobre la autenticación en AWS IoT: [Sumérgete en AWS IoT Autenticación y autorización](#).

Información general del certificado X.509

Los certificados X.509 son certificados digitales que utilizan el [estándar de infraestructura de clave pública X.509](#) para asociar una clave pública a una identidad contenida en un certificado. Los certificados X.509 se generan a través de una entidad de confianza conocida como autoridad de certificación (CA). La CA administra uno o varios certificados especiales llamados certificados de CA, que utiliza para generar certificados X.509. Solo la autoridad de certificación tiene acceso a los certificados de CA. Las cadenas de certificados X.509 se utilizan tanto para la autenticación del servidor por parte de los clientes como para la autenticación del cliente por parte del servidor.

Autenticación del servidor

Cuando el dispositivo u otro cliente intenta conectarse a AWS IoT Core, el servidor de AWS IoT Core enviará un certificado X.509 que el dispositivo utiliza para autenticar el servidor. La autenticación se lleva a cabo en la capa TLS mediante la validación de la [cadena de certificados X.509 \(p. 298\)](#). Este es el mismo método que utiliza el navegador cuando visita una URL HTTPS. Si desea utilizar certificados de su propia autoridad de certificación, consulte [Administración de sus certificados de entidad de certificación \(p. 302\)](#).

Cuando sus dispositivos u otros clientes establecen una conexión TLS con un punto de enlace de AWS IoT Core, AWS IoT Core presenta una cadena de certificados que los dispositivos utilizan para verificar si se están comunicando con AWS IoT Core y no con otro servidor que suplante a AWS IoT Core. La cadena que se presenta depende de una combinación del tipo de punto de enlace al que se conecta el dispositivo y del [conjunto de cifrado \(p. 382\)](#) que el cliente y AWS IoT Core negociaron durante el protocolo TLS.

Tipo de punto de enlace

AWS IoT Core admite dos tipos diferentes de puntos de enlace de datos: `iot:Data` e `iot:Data-ATS`. Los puntos de enlace `iot:Data` presentan un certificado firmado por el [certificado de entidad de certificación raíz de VeriSign Class 3 Public Primary G5](#). Los puntos de enlace `iot:Data-ATS` presentan un certificado de servidor firmado por una entidad emisora de certificados de [Amazon Trust Services](#).

Los certificados presentados por los puntos de enlace de ATS están firmados por Starfield. Algunas implementaciones de cliente TLS requieren la validación de la raíz de confianza y requieren que los certificados de CA de Starfield estén instalados en los almacenes de confianza del cliente.

Warning

No se recomienda utilizar un método de fijación de certificados que aplica hash en todo el certificado (incluido el nombre del emisor, etc.) porque esto provocará un error en la verificación del certificado porque los certificados ATS que proporcionamos están firmados de forma cruzada por Starfield y tienen un nombre de emisor diferente.

Use puntos de enlace `iot:Data-ATS` a menos que su dispositivo requiera certificados de CA de Symantec o Verisign. Los certificados de Symantec y Verisign han quedado obsoletos y ya no son compatibles con la mayoría de los navegadores web.

Puede utilizar el comando `describe-endpoint` para crear el punto de enlace de ATS.

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

El comando `describe-endpoint` devuelve un punto de enlace en el formato siguiente.

```
account-specific-prefix.iot.your-region.amazonaws.com
```

La primera vez que se llama a `describe-endpoint`, se crea un punto de enlace. Todas las llamadas posteriores a `describe-endpoint` devuelven el mismo punto de enlace.

En lo que se refiere a la compatibilidad con versiones anteriores, AWS IoT Core sigue siendo compatible con los puntos de enlace de Symantec. Para obtener más información, consulte [How AWS IoT Core is Helping Customers Navigate the Upcoming Distrust of Symantec Certificate Authorities](#). Los dispositivos que funcionan en los puntos de enlace de ATS son totalmente interoperables con los dispositivos que funcionan en los puntos de enlace de Symantec en la misma cuenta y no es necesario volver a registrarlos.

Note

Para ver el punto de enlace `iot:Data-ATS` en la consola de AWS IoT Core, elija **Settings** (**Configuración**). La consola solo muestra el punto de enlace `iot:Data-ATS`. De forma predeterminada, el comando `describe-endpoint` muestra el punto de enlace `iot:Data` para garantizar la compatibilidad con versiones anteriores. Para ver el punto de enlace `iot:Data-ATS`, especifique el parámetro `--endpointType` como en el ejemplo anterior.

Crear un `IotDataPlaneClient` con el AWS SDK para Java

De forma predeterminada, el [AWS SDK para Java - Versión 2](#) crea un `IotDataPlaneClient` utilizando un punto de enlace `iot:Data`. Para crear un cliente que utilice un punto de enlace `iot:Data-ATS`, debe hacer lo siguiente.

- Cree un punto de enlace `iot:Data-ATS` utilizando la API `DescribeEndpoint`.
- Especifique ese punto de enlace al crear el `IotDataPlaneClient`.

En el ejemplo siguiente se realizan ambas operaciones.

```
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();
    IotDataPlaneClient iot = IotDataPlaneClient.builder()
        .credentialsProvider(CREDENTIALS_PROVIDER_CHAIN)
        .endpointOverride(URI.create("https://" + endpoint))
        .region(Region.US_EAST_1)
        .build();
}
```

Certificados de entidad de certificación para autenticación de servidor

En función del tipo de punto de enlace de datos que esté utilizando y del conjunto de cifrado que haya negociado, los certificados de autenticación del servidor de AWS IoT Core estarán firmados por uno de los siguientes certificados de la entidad de certificación raíz:

Puntos de enlace de VeriSign (heredados)

- Clave RSA de 2048 bits: [Certificado de CA raíz G5 principal y público de clase 3 de VeriSign](#)

Puntos de enlace de Amazon Trust Services (preferidos)

Note

Es posible que tenga que hacer clic con el botón derecho en estos enlaces y seleccionar Guardar enlace como... para guardar estos certificados como archivos.

- Clave RSA de 2048 bits:[Amazon Root CA 1](#).
- Clave RSA de 4096 bits: Amazon Root CA 2. Reservada para futura utilización.
- Clave ECC de 256 bits:[Amazon Root CA 3](#).

- Clave ECC de 384 bits: Amazon Root CA 4. Reservada para futura utilización.

Todos estos certificados tienen firma cruzada del [Certificado Starfield Root CA](#). A partir del lanzamiento de AWS IoT Core en la región de Asia Pacífico (Mumbai) el 9 de mayo de 2018, todas las regiones nuevas de AWS IoT Core proporcionan exclusivamente certificados de ATS.

Diretrices de autenticación de servidores

Hay muchas variables que pueden afectar a la capacidad de un dispositivo para validar el certificado de autenticación del servidor de AWS IoT Core. Por ejemplo, los dispositivos pueden tener demasiada memoria limitada para contener todos los certificados de CA raíz posibles, o los dispositivos pueden implementar un método no estándar de validación de certificados. Por estas razones, sugerimos seguir estas directrices:

- Le recomendamos que utilice su punto de enlace ATS e instale todos los admitidos [Amazon Root CA certificados](#).
- Si no puede almacenar todos estos certificados en su dispositivo y si sus dispositivos no utilizan la validación basada en ECC, puede omitir el [Amazon Root CA 3](#) y [Amazon Root CA 4](#) Certificados ECC. Si sus dispositivos no implementan una validación de certificados basada en RSA, puede omitir la [Amazon Root CA 1](#) y [Amazon Root CA 2](#) Certificados RSA. Es posible que tenga que hacer clic con el botón derecho en estos enlaces y seleccionar Guardar enlace como... para guardar estos certificados como archivos.
- Si tiene problemas de validación de certificados de servidor al conectarse a su punto de enlace de ATS, intente agregar el certificado Amazon Root CA correspondiente con firma cruzada a su almacén de confianza. Es posible que tenga que hacer clic con el botón derecho en estos enlaces y seleccionar Guardar enlace como... para guardar estos certificados como archivos.
 - Firmado cruzado [Amazon Root CA 1](#)
 - Firmado cruzado [Amazon Root CA 2](#)- Reservada para future utilización.
 - Firmado cruzado [Amazon Root CA 3](#)
 - Firmado cruzado [Amazon Root CA 4](#)- Reservada para future utilización.
- Si experimenta problemas de validación de certificados de servidor, es posible que el dispositivo deba confiar explícitamente en la CA raíz. Pruebe a agregar la [Starfield Root CA Certificate](#) a tu tienda de confianza.
- Si sigue teniendo problemas después de ejecutar los pasos anteriores, póngase en contacto con [AWSDeveloper Support](#).

Note

Los certificados de CA tienen una fecha de vencimiento posterior que no pueden usar para validar un certificado del servidor. Los certificados de CA podrían tener que reemplazarse antes de su fecha de vencimiento. Asegúrese de que puede actualizar los certificados de entidad de certificación raíz en todos sus dispositivos o clientes para asegurarse de que la conectividad se mantenga y esté al día de las prácticas recomendadas de seguridad.

Note

Cuando se conecte a AWS IoT Core en el código del dispositivo, pase el certificado a la API que está utilizando para realizar la conexión. La API que use variará según el SDK. Para obtener más información, consulte los [SDK de dispositivo de AWS IoT Core \(p. 1274\)](#).

Autenticación del cliente

AWS IoT admite tres tipos de entidades principales de identidad para la autenticación de dispositivos o clientes:

- Certificados de cliente X.509 ([p. 298](#))
- Usuarios, grupos y roles de IAM ([p. 318](#))
- Identidades de Amazon Cognito ([p. 319](#))

Estas identidades se pueden usar con dispositivos, aplicaciones móviles, web o de escritorio. Incluso los puede utilizar un usuario que escribe comandos de interfaz de línea de comandos (CLI) de AWS IoT. Por lo general, los dispositivos utilizan certificados X.509, mientras que las aplicaciones móviles utilizan identidades de Amazon Cognito. Las aplicaciones web y de escritorio usan IAM o identidades federadas. Los comandos AWS CLI usan IAM. Para obtener más información acerca de identidades de IAM, consulte [Identity and Access Management en AWS IoT \(p. 384\)](#).

Certificados de cliente X.509

Los certificados X.509 proporcionan a AWS IoT la capacidad de autenticar las conexiones de clientes y dispositivos. Los certificados de cliente deben registrarse con AWS IoT antes de que un cliente pueda comunicarse con AWS IoT. Un certificado de cliente se puede registrar en varios cuentas de AWS en la misma región para facilitar el movimiento de dispositivos entre sus cuentas. Para obtener más información, consulte [Uso de certificados de cliente X.509 en varias cuentas \(p. 299\)](#).

Se recomienda que cada dispositivo o cliente reciba un certificado único para permitir acciones de administración de clientes precisas, incluida la revocación de certificados. Los dispositivos y los clientes deben ser compatibles con la rotación y la sustitución de certificados para garantizar un buen funcionamiento cuando los certificados caduquen.

Para obtener información sobre el uso de certificados X.509 para admitir más de unos pocos dispositivos, consulte [Aprovisionamiento de dispositivos \(p. 795\)](#) para revisar las distintas opciones de aprovisionamiento y administración de certificados que admite AWS IoT.

AWS IoT admite estos tipos de certificados de cliente X.509:

- Certificados X.509 generados por AWS IoT
- Certificados X.509 firmados por una entidad de certificación registrada con AWS IoT.
- Certificados X.509 firmados por una entidad de certificación que no está registrada con AWS IoT.

En esta sección se describe cómo administrar certificados X.509 en AWS IoT. Puede utilizar la consola de AWS IoT o la AWS CLI para realizar estas operaciones de certificado:

- [Crear certificados de cliente de AWS IoT \(p. 300\)](#)
- [Creación de sus propios certificados de cliente \(p. 301\)](#)
- [Registrar un certificado de cliente \(p. 308\)](#)
- [Activar o desactivar un certificado de cliente \(p. 312\)](#)
- [Revocar un certificado de cliente \(p. 314\)](#)

Para obtener más información sobre los comandos de la AWS CLI que realizan estas operaciones, consulte la [Referencia de la CLI de AWS IoT](#).

Uso de certificados de cliente X.509

Los certificados X.509 autentican las conexiones de cliente y dispositivo en AWS IoT. Los certificados X.509 ofrecen varios beneficios con respecto a otros mecanismos de identificación y autenticación. Los certificados X.509 permiten usar claves asimétricas con los dispositivos. Por ejemplo, podría forzar claves privadas en un almacenamiento seguro en un dispositivo para que el material criptográfico confidencial

nunca salga del dispositivo. Los certificados X.509 proporcionan una autenticación del cliente más fiable que los otros sistemas, como el nombre de usuario y la contraseña o los tokens de portador, ya que la clave privada jamás abandona el dispositivo.

AWS IoT autentica los certificados utilizando el modo de autenticación de cliente del protocolo TLS. La compatibilidad con TLS está disponible en numerosos lenguajes de programación y sistemas operativos, y se utiliza generalmente para cifrar datos. En la autenticación de cliente de TLS, AWS IoT solicita un certificado X.509 de cliente y valida el estado del certificado y la cuenta de AWS contra un registro de certificados. A continuación, desafía al cliente para obtener una prueba de propiedad de la clave privada que corresponde a la clave pública contenida en el certificado. AWS IoT requiere que los clientes envíen la [extensión de Indicación de nombre de servidor \(SNI\)](#) al protocolo de Transport Layer Security (TLS). Para obtener más información sobre la configuración de la extensión SNI, consulte [Seguridad de transporte en AWS IoT \(p. 382\)](#).

Los certificados X.509 se pueden verificar con una entidad de certificación (CA) de confianza. Puede crear certificados de cliente que utilicen la entidad de certificación Amazon Root y puede utilizar sus propios certificados de cliente firmados por otra entidad de certificación. Para obtener más información sobre el uso de sus propios certificados X.509, consulte [Creación de sus propios certificados de cliente \(p. 301\)](#).

La fecha y hora de caducidad de los certificados firmados por un certificado de entidad de certificación se establecen en el momento de su creación. Los certificados X.509 generados por AWS IoT caducan a medianoche UTC del 31 de diciembre de 2049 (2049-12-31T23:59:59Z). Para obtener más información sobre el uso de la consola de AWS IoT para crear certificados que utilicen la entidad de certificación de Amazon Root, consulte [Crear certificados de cliente de AWS IoT \(p. 300\)](#).

Uso de certificados de cliente X.509 en variosCuenta de AWSs con registro de varias cuentas

El registro de varias cuentas permite mover dispositivos entre suCuenta de AWSs en la misma región o en regiones diferentes. Con esto, puede registrar, probar y configurar un dispositivo en una cuenta de preproducción y, a continuación, registrar y utilizar el mismo dispositivo y certificado de dispositivo en una cuenta de producción. También puede [registrar el certificado de cliente en el dispositivo \(los certificados de dispositivo\) sin una entidad de certificación \(p. 310\)](#) registrada con AWS IoT.

Note

Los certificados utilizados para el registro de varias cuentas se admiten en `eliot:Data-ATS,iot:Data(heredado),iot:Jobs,yiot:CredentialProvider` tipos de punto de enlace. Para obtener más información acerca de AWS IoT puntos de enlace de dispositivo, consulte [AWS IoT datos de dispositivos y puntos finales de servicio \(p. 78\)](#).

Note

El registro de varias cuentas no admite just-in-time registro porque se requiere un certificado de verificación para registrar la CA, que solo se genera para una cuenta específica.

Los dispositivos que utilizan el registro de varias cuentas deben enviar el [Extensión de indicación de nombre de servidor \(SNI\)](#) al protocolo Transport Layer Security (TLS) y proporcione la dirección completa del punto de enlace en `host_name`, cuando se conectan a AWS IoT. AWS IoT Utiliza la dirección del endpoint `host_name` para enrutar la conexión a la correcta AWS IoT account. Los dispositivos existentes que no envíen una dirección de punto de enlace válida en `host_name` seguirán funcionando, pero no podrán utilizar las características que requiere esta información. Para obtener más información acerca de la extensión SNI y para aprender a identificar la dirección del punto de enlace del campo `host_name`, consulte [Seguridad de transporte en AWS IoT \(p. 382\)](#).

Para utilizar el registro de varias cuentas

1. No registre la entidad de certificación con la que firmó los certificados del dispositivo con AWS IoT.

2. Registre los certificados de dispositivo sin una entidad de certificación. Consulte [Registrar un certificado de cliente firmado por una entidad de certificación no registrada \(CLI\) \(p. 310\)](#).
3. Utilice el `host_name` correcto en la extensión SNI para TLS cuando el dispositivo se conecte a AWS IoT. Consulte [Seguridad de transporte en AWS IoT \(p. 382\)](#).

Algoritmos de firma de certificados admitidos por AWS IoT

AWS IoT es compatible con los siguientes algoritmos de firma de certificado:

- SHA256WITHRSA
- SHA384WITHRSA
- SHA512WITHRSA
- DSA_WITH_SHA256
- ECDSA-WITH-SHA256
- ECDSA-WITH-SHA384
- ECDSA-WITH-SHA512

Crear certificados de cliente de AWS IoT

AWS IoT proporciona certificados de cliente firmados por la entidad de certificación (CA) Amazon Root.

En este tema se describe cómo crear un certificado de cliente firmado por la entidad de certificación Amazon Root y descargar los archivos de certificado. Después de crear los archivos de certificado de cliente, debe instalarlos en el cliente.

Note

Cada certificado de cliente X.509 proporcionado por AWS IoT contiene los atributos de emisor y asunto que se establecen en el momento de la creación del certificado. Los atributos del certificado son inmutables solo después de crear el certificado.

Puede utilizar la consola de AWS IoT o la AWS CLI para crear un certificado de AWS IoT firmado por la entidad de certificación Amazon Root.

Crear un certificado de AWS IoT (consola)

Para crear un certificado de AWS IoT mediante la consola de AWS IoT

1. Inicie sesión en AWS La consola de administración de y abra la [AWS IoT consola](#).
2. En el panel de navegación izquierdo, elija Security (Seguridad), elija Certificates (Certificados) y, a continuación, elija Create (Crear).
3. Elija One-click certificate creation (recommended) [Creación de un certificado con un clic (recomendado)] - Create certificate (Crear certificado).
4. En la página Certificate created! (!Certificado creado!), descargue los archivos de certificado del cliente para el objeto, clave pública y clave privada en una ubicación segura. Estos certificados generados por AWS IoT solo están disponibles para su uso con AWS IoT Servicios de .

Si necesita además el archivo de certificado de entidad de certificación de Amazon Root, esta página también tiene el enlace a la página desde la que puede descargarlo.

5. Ahora se ha creado y registrado un certificado de cliente con AWS IoT. Debe activar el certificado antes de usarlo en un cliente.

ElegirActivarPara activar el certificado de cliente ahora. Si no desea activar el certificado ahora, [Activar un certificado de cliente \(consola\) \(p. 312\)](#) describe cómo activarlo más adelante.

6. Si desea asociar una política al certificado, elija Asociar una política.

Si no desea asociar una política ahora, elija Terminar para terminar. Puede asociar una política más adelante.

Después de completar el procedimiento, instale los archivos de certificado en el cliente.

Crear un certificado (CLI) de AWS IoT

La AWS CLI proporciona el comando [create-keys-and-certificate](#) para crear certificados de cliente firmados por la entidad emisora de certificados de Amazon Root. No obstante, este comando no descarga el archivo de certificado de entidad de certificación de Amazon Root. Puede descargar el archivo de certificado de entidad de certificación de Amazon Root desde [Certificados de entidad de certificación para autenticación de servidor \(p. 296\)](#).

Este comando crea archivos de clave privada, clave pública y certificado X.509 y registra y activa el certificado con 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
```

Si no desea activar el certificado al crearlo y registrararlo, este comando crea archivos de clave privada, clave pública y certificado X.509 y registra el certificado, pero no lo activa. [Activar un certificado de cliente \(CLI\) \(p. 313\)](#) describe cómo activar el certificado más adelante.

```
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 los archivos de certificado en el cliente.

Creación de sus propios certificados de cliente

AWS IoT admite certificados de cliente firmados por otras entidades de certificados raíz (entidades de certificación). Puede registrar certificados de cliente firmados por otra entidad de certificación raíz; sin embargo, si desea que el dispositivo o cliente registre su certificado de cliente cuando se conecte por primera vez AWS IoT, la entidad de certificación raíz debe estar registrada con AWS IoT.

Note

Un certificado de entidad de certificación solo se puede registrar mediante una cuenta en una región.

Para obtener más información acerca del uso de certificados X.509 para admitir más de unos pocos dispositivos, consulte [Aprovisionamiento de dispositivos \(p. 795\)](#) para revisar las diferentes opciones de administración y aprovisionamiento de certificados que admite AWS IoT.

Temas

- [Administración de sus certificados de entidad de certificación \(p. 302\)](#)
- [Creación de un certificado de cliente mediante el certificado de entidad de certificación \(p. 307\)](#)

Administración de sus certificados de entidad de certificación

En esta sección se describen las tareas comunes para administrar sus propios certificados de entidad de certificación.

Es posible que tenga que registrar la entidad de certificación con AWS IoT si está utilizando certificados de cliente firmados por una entidad emisora de certificados que AWS IoT no reconoce.

Si desea que los clientes registren automáticamente sus certificados de cliente con AWS IoT la primera vez que se conectan, la entidad de certificación que firmó los certificados de cliente debe estar registrada con AWS IoT. De lo contrario, no es necesario registrar el certificado de entidad de certificación que firmó los certificados de cliente.

Note

Un certificado de entidad de certificación solo se puede registrar mediante una cuenta en una región.

- [Creación de un certificado de entidad de certificación \(p. 302\)](#), si necesita uno.

Cree los archivos de certificado y clave que necesita para el siguiente paso.

- [Registro de su certificado de entidad de certificación \(p. 302\)](#)

Registre su certificado de entidad de certificación con AWS IoT

- [Desactivar un certificado de entidad de certificación \(p. 306\)](#)

Creación de un certificado de entidad de certificación

Si no dispone de certificado de entidad de certificación, puede utilizar [OpenSSL v1.1](#) herramientas para crear una.

Note

No puede realizar este procedimiento en la consola de AWS IoT.

Para crear un certificado de entidad de certificación mediante [OpenSSL v1.1](#) herramientas

1. Genere un par de claves.

```
openssl genrsa -out root_CA_key_filename.key 2048
```

2. Utilice la clave privada del par de claves para generar un certificado de CA.

```
openssl req -x509 -new -nodes \
-key root_CA_key_filename.key \
-sha256 -days 1024 \
-out root_CA_cert_filename.pem
```

Registro de su certificado de entidad de certificación

Estos procedimientos registran un certificado de entidad de certificación de AWS IoT.

Registro de un certificado de entidad de certificación (consola).

Note

Para registrar un certificado de CA en la consola, inicie en la consola en[Registro de un certificado de entidad](#).

Registro de un certificado de entidad de certificación (CLI)

Asegúrese de que dispone de lo siguiente en su equipo antes de continuar:

- El archivo de certificado de la CA raíz (al que se hace referencia a continuación como`root_CA_cert_filename.pem`)
- El archivo de clave privada del certificado de CA raíz (al que se hace referencia a continuación como`root_CA_key_filename.key`)
- `OpenSSL v1.1`o posterior

Para registrar un certificado de entidad de certificación mediante la AWS CLI

1. Utilice `get-registration-code` para obtener un código de registro de AWS IoT. Guarde el `registrationCode` devuelto para usarlo como Common Name del certificado de verificación de clave privada.

```
aws iot get-registration-code
```

2. Genere un par de claves para el certificado de verificación de clave privada:

```
openssl genrsa -out verification_cert_key_filename.key 2048
```

3. Cree una solicitud de firma de certificado (CSR) para el certificado de verificación de clave privada. Establezca el campo Common Name del certificado en el `registrationCode` devuelto por `get-registration-code`.

```
openssl req -new \
    -key verification_cert_key_filename.key \
    -out verification_cert_csr_filename.csr
```

Se le solicita información, incluido el Common Name del 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. Utilice la CSR para crear un certificado de verificación de la clave 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 el certificado de CA en AWS IoT. Pase el nombre de archivo del certificado de entidad de certificación y el nombre de archivo del certificado de verificación de clave privada al comando `register-ca-certificate`:

```
aws iot register-ca-certificate \
    --ca-certificate file://root_CA_cert_filename.pem \
    --verification-cert file://verification_cert_filename.pem
```

Este comando devuelve el `certificateID`, si se realiza correctamente.

6. En este punto, el certificado de entidad de certificación se ha registrado con AWS IoT, pero no está activo. El certificado de entidad de certificación debe estar activo antes de poder registrar los certificados de cliente firmados por él.

Este paso activa el certificado de entidad de certificación.

Ejecute el comando de CLI `update-certificate` para activar el certificado de entidad de certificación:

```
aws iot update-ca-certificate \
    --certificate-id certificateId \
    --new-status ACTIVE
```

Utilice el comando `describe-ca-certificate` para ver el estado del certificado de entidad de certificación.

Crear un certificado de verificación de CA para registrar el certificado de CA en la consola

Note

Este procedimiento solo se utiliza si está registrando un certificado de CA desde el AWS IoTconsola de .

Si no acudió a este procedimiento desde elAWS IoTconsola, inicie el proceso de registro de certificados de CA en la consola en[Registro de un certificado de entidad](#).

Asegúrese de que dispone de lo siguiente en el mismo equipo antes de continuar:

- El archivo de certificado de la CA raíz (al que se hace referencia a continuación como`root_CA_cert_filename.pem`)
- El archivo de clave privada del certificado de CA raíz (al que se hace referencia a continuación como`root_CA_key_filename.key`)
- `OpenSSL v1.1` o posterior

Para utilizar la interfaz de línea de comandos para crear un certificado de verificación de CA para registrar el certificado de CA en la consola

1. Reemplazar`verification_cert_key_filename.key`con el nombre del archivo de clave de certificado de verificación que desea crear. Por ejemplo,`verification_cert.key`, a continuación, ejecute este comando de para generar un key pair para el certificado de verificación de clave privada:

```
openssl genrsa -out verification_cert_key_filename.key 2048
```

2. Reemplazar `verification_cert_key_filename.key` con el nombre del archivo clave que creó en el paso 1.

Reemplazar `verification_cert_csr_filename.csr` con el nombre del archivo de solicitud de firma de certificado (CSR) que desea crear. Por ejemplo, `verification_cert.csr`.

Ejecute este comando para crear el archivo CSR.

```
openssl req -new \
-key verification_cert_key_filename.key \
-out verification_cert_csr_filename.csr
```

El comando le solicita información adicional que se explica más adelante.

3. En el navegador AWS IoT Consola de, en Certificado de verificación contenedor, copia el código de registro.
4. La información que el OpenSSL muestra en el siguiente ejemplo. Except la Common Name, puede introducir sus propios valores o dejarlos en blanco.

En el navegador Common Name, pegue el código de registro que copió en el paso 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 []:
```

Una vez finalizado, el comando crea el archivo CSR.

5. Reemplazar `verification_cert_csr_filename.csr` con `verification_cert_csr_filename.csr` que utilizó en el paso anterior.

Reemplazar `root_CA_cert_filename.pem` con el nombre de archivo del certificado de CA que desea registrar.

Reemplazar `root_CA_key_filename.key` con el nombre de archivo del archivo de clave privada del certificado de CA.

Reemplazar `verification_cert_filename.pem` con el nombre de archivo del certificado de verificación que desea crear. Por ejemplo, `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. Una vez finalizado el comando openssl, debe tener estos archivos listos para usar cuando vuelva a la consola.
 - El archivo de certificado de CA (*root_CA_cert_filename.pem* utilizado en el comando anterior)
 - El certificado de verificación que creó en el paso anterior (*verification_cert_filename.pem* utilizado en el comando anterior)

Desactivar un certificado de entidad de certificación

Cuando un certificado de entidad de certificación está habilitado para el registro automático de certificados de cliente, AWS IoT comprueba el certificado de entidad de certificación utilizado para firmar el certificado de cliente para asegurarse de que la entidad de certificación esté ACTIVE. Si el certificado de entidad de certificación es INACTIVE, AWS IoT no permite que se registre el certificado de cliente.

Al establecer el certificado de entidad de certificación como INACTIVE, impide que los certificados de cliente nuevos emitidos por la entidad de certificación se registren automáticamente.

Note

Todos los certificados de dispositivo registrados que haya firmado el certificado de entidad de certificación en riesgo siguen funcionando hasta que revoque explícitamente cada uno de ellos.

Desactivar un certificado de entidad de certificación (consola)

Para desactivar un certificado de entidad de certificación mediante la consola de AWS IoT

1. Inicie sesión en AWS La consola de administración de y abra la [AWS IoT consola](#).
2. En el panel de navegación de la izquierda, elija Secure (Seguridad) y, a continuación, elija CAs (Entidades de certificación).
3. En la lista de entidades de certificación, busque la que desea desactivar y abra el menú de opciones mediante el icono de puntos suspensivos.
4. En el menú de opciones, elija Deactivate (Desactivar).

La entidad de certificación debe mostrarse como Inactive (Inactiva) en la lista.

Note

La consola de AWS IoT no proporciona una forma de enumerar los certificados firmados por la entidad de certificación desactivada. Para obtener una opción de AWS CLI para enumerar esos certificados, consulte [Desactivar un certificado de entidad de certificación \(CLI\) \(p. 306\)](#).

Desactivar un certificado de entidad de certificación (CLI)

La AWS CLI proporciona el comando `update-ca-certificate` para desactivar un certificado de entidad de certificación.

```
aws iot update-ca-certificate \
--certificate-id certificateId \
--new-status INACTIVE
```

Utilice el comando `list-certificates-by-ca` para obtener una lista de todos los certificados de cliente registrados firmados por la entidad de certificación especificada. Por cada certificado de cliente firmado

por el certificado de entidad de certificación especificado, puede utilizar el comando [update-certificate](#) para revocar el certificado de cliente y evitar que este se use.

Utilice el comando [describe-ca-certificate](#) para ver el estado del certificado de entidad de certificación.

Creación de un certificado de cliente mediante el certificado de entidad de certificación

Puede utilizar su propia entidad de certificación (CA) para crear certificados de cliente. El certificado de cliente debe registrarse en AWS IoT para poder utilizarlo. Para obtener información acerca de las opciones de registro de los certificados de cliente, consulte [Registrar un certificado de cliente \(p. 308\)](#).

Crear un certificado de cliente (CLI)

Note

No puede realizar este procedimiento en la consola de AWS IoT.

Para crear un certificado de cliente mediante la AWS CLI

1. Genere un par de claves.

```
openssl genrsa -out device_cert_key_filename.key 2048
```

2. Cree una CSR para el certificado de cliente.

```
openssl req -new \
    -key device_cert_key_filename.key \
    -out device_cert_csr_filename.csr
```

Se le solicita que indique información, tal y como se muestra aquí:

```
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. Cree un certificado de cliente a partir de la 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
```

En este punto, se ha creado el certificado de cliente, pero aún no se ha registrado con AWS IoT. Para obtener información acerca de cómo y cuándo registrar el certificado de cliente, consulte [Registrar un certificado de cliente \(p. 308\)](#).

Registrar un certificado de cliente

Los certificados de cliente deben registrarse con AWS IoT para habilitar las comunicaciones entre el cliente y AWS IoT. Puede registrar cada certificado de cliente manualmente o puede configurar los certificados de cliente para que se registren automáticamente cuando el cliente se conecte a AWS IoT por primera vez.

Si desea que sus clientes y dispositivos registren sus certificados de cliente cuando se conectan por primera vez, debe usar [Registro de su certificado de entidad de certificación \(p. 302\)](#) para firmar el certificado de cliente con AWS IoT en las regiones en las que desea usarlo. La entidad de certificación de Amazon Root se registra automáticamente en AWS IoT.

Los certificados de cliente pueden compartirse conCuenta de AWSs y regiones. Los procedimientos de estos temas deben realizarse en cada cuenta y región en la que desee utilizar el certificado de cliente. El registro de un certificado de cliente en una cuenta o región no es reconocido automáticamente por otra.

Note

Los clientes que utilizan el protocolo Transport Layer Security (TLS) para conectarse a AWS IoT deben admitir la [extensión de indicación de nombre de servidor \(SNI\)](#) para TLS. Para obtener más información, consulte [Seguridad de transporte en AWS IoT \(p. 382\)](#).

Temas

- [Registrar manualmente un certificado de cliente \(p. 308\)](#)
- [Registrar un certificado de cliente cuando el cliente se conecta aAWS IoT just-in-time registro \(JITR\) \(p. 311\)](#)

Registrar manualmente un certificado de cliente

Puede registrar un certificado de cliente manualmente mediante la consola de AWS IoT y la AWS CLI.

El procedimiento de registro que se va a utilizar depende de si el certificado será compartido porCuenta de AWSs y regiones. El registro de un certificado de cliente en una cuenta o región no es reconocido automáticamente por otra.

Los procedimientos de este tema deben realizarse en cada cuenta y región en la que desee utilizar el certificado de cliente. Los certificados de cliente pueden compartirse conCuenta de AWSs y regiones, pero solo si el certificado de cliente está firmado por una entidad de certificación (CA) que NO esté registrada enAWS IoT.

Registrar un certificado de cliente firmado por una entidad de certificación registrada (consola)

Note

Antes de realizar este procedimiento, asegúrese de que tiene el archivo.pem del certificado de cliente y de que el certificado de cliente fue firmado por una entidad de certificación [que se ha registrado con AWS IoT \(p. 302\)](#).

Para registrar un certificado existente con el AWS IoT mediante la consola

1. Inicie sesión enAWSLa consola de administración de y abra la[AWS IoTconsola](#).
2. En el panel de navegación izquierdo, elija Security (Seguridad), elija Certificates (Certificados) y, a continuación, elija Create (Crear).
3. SobreCrear un certificado, localice elUtilizar mi certificadoentrada y elijalntroducción.

4. En Select a CA (Seleccionar una entidad de certificación):
 - Si los certificados de cliente están firmados por una entidad de certificación registrada con AWS IoT
Elija esa entidad de certificación de la lista y, a continuación, elija Próximo.
 - Si los certificados de cliente no están firmados por una entidad de certificación registrada con AWS IoT
Consulte [Registrar un certificado de cliente firmado por una entidad de certificación no registrada \(consola\) \(p. 309\)](#).
 - Si los certificados de cliente están firmados por la entidad de certificación de Amazon
No seleccione ninguna entidad de certificación, solo elija Próximo.

Si los certificados de cliente no están firmados por una entidad de certificación registrada con AWS IoT, consulte [Registrar un certificado de cliente firmado por una entidad de certificación no registrada \(consola\) \(p. 309\)](#).

5. Sobre Registrar certificados de dispositivo existentes, elige Seleccionar certificados y seleccione hasta 10 archivos de certificado para registrarse.
6. Despues de cerrar el cuadro de diálogo de archivo, seleccione si desea activar o revocar los certificados de cliente cuando los registre.

Si no activa un certificado cuando se registra, [Activar un certificado de cliente \(consola\) \(p. 312\)](#) describe cómo activarlo más adelante.

Si un certificado se revoca cuando se registra, no se puede activar más tarde.

Después de elegir los archivos de certificado que desea registrar y seleccionar las acciones que desea realizar después del registro, seleccione Registrar certificados.

Los certificados de cliente registrados correctamente aparecen en la lista de certificados.

[Registrar un certificado de cliente firmado por una entidad de certificación no registrada \(consola\)](#)

Note

Antes de realizar este procedimiento, asegúrese de que tiene el archivo .pem del certificado de cliente.

Para registrar un certificado existente con el AWS IoT mediante la consola

1. Inicie sesión en AWS La consola de administración de y abra la [AWS IoT consola](#).
2. En el panel de navegación izquierdo, elija Security (Seguridad), elija Certificates (Certificados) y, a continuación, elija Create (Crear).
3. Sobre Crear un certificado, localice el Utilizar mi certificado entrada y elija Introducción.
4. Sobre Seleccione una CA, elige Próximo.
5. Sobre Registrar certificados de dispositivo existentes, elige Seleccionar certificados y seleccione hasta 10 archivos de certificado para registrarse.
6. Despues de cerrar el cuadro de diálogo de archivo, seleccione si desea activar o revocar los certificados de cliente cuando los registre.

Si no activa un certificado cuando se registra, [Activar un certificado de cliente \(consola\) \(p. 312\)](#) describe cómo activarlo más adelante.

Si un certificado se revoca cuando se registra, no se puede activar más tarde.

Después de elegir los archivos de certificado que desea registrar y seleccionar las acciones que desea realizar después del registro, elija Registrar certificados.

Los certificados de cliente registrados correctamente aparecen en la lista de certificados.

Registrar un certificado de cliente firmado por una entidad de certificación registrada (CLI)

Note

Antes de realizar este procedimiento, asegúrese de que tiene el archivo .pem de la entidad de certificación y el archivo .pem del certificado de cliente. El certificado de cliente debe estar firmado por una entidad de certificación que haya [registrado con AWS IoT \(p. 302\)](#).

Utilice el comando [register-certificate](#) para registrar, pero no activar, un certificado de cliente.

```
aws iot register-certificate \
--certificate-pem file://device_cert_filename.pem \
--ca-certificate-pem file://ca_cert_filename.pem
```

El certificado de cliente está registrado con AWS IoT, pero no está activo aún. Consulte [Activar un certificado de cliente \(CLI\) \(p. 313\)](#) para obtener información sobre cómo activarlo más adelante.

También puede activar el certificado de cliente cuando lo registre utilizando este comando.

```
aws iot register-certificate \
--set-as-active \
--certificate-pem file://device_cert_filename.pem \
--ca-certificate-pem file://ca_cert_filename.pem
```

Para obtener más información acerca de cómo activar el certificado para que se pueda utilizar para conectarse a AWS IoT, consulte [Activar o desactivar un certificado de cliente \(p. 312\)](#).

Registrar un certificado de cliente firmado por una entidad de certificación no registrada (CLI)

Note

Antes de llevar a cabo este procedimiento, asegúrese de que tiene el archivo .pem del certificado.

Utilice el comando [register-certificate-without-ca](#) para registrar, pero no activar, un certificado de cliente.

```
aws iot register-certificate-without-ca \
--certificate-pem file://device_cert_filename.pem
```

El certificado de cliente está registrado con AWS IoT, pero no está activo aún. Consulte [Activar un certificado de cliente \(CLI\) \(p. 313\)](#) para obtener información sobre cómo activarlo más adelante.

También puede activar el certificado de cliente cuando lo registre utilizando este comando.

```
aws iot register-certificate-without-ca \
--status ACTIVE \
--certificate-pem file://device_cert_filename.pem
```

Para obtener más información acerca de cómo activar el certificado de forma que se pueda utilizar para conectarse a AWS IoT, consulte [Activar o desactivar un certificado de cliente \(p. 312\)](#).

Registrar un certificado de cliente cuando el cliente se conecta aAWS IoT just-in-time registro (JITR)

Puede configurar un certificado de entidad de certificación para habilitar los certificados de cliente con los que ha firmado para que se registren automáticamente con AWS IoT la primera vez que el cliente se conecta a AWS IoT.

Para registrar certificados de cliente cuando un cliente se conecta a AWS IoT por primera vez, debe habilitar el certificado de entidad de certificación para el registro automático y configurar la primera conexión del cliente para proporcionar los certificados necesarios.

Configurar un certificado de entidad de certificación para admitir el registro automático (consola)

Para configurar un certificado de entidad de certificación para admitir el registro automático de certificados de cliente mediante la consola de AWS IoT

1. Inicie sesión enAWSLa consola de administración de y abra la[AWS IoTconsola](#).
2. En el panel de navegación de la izquierda, elija Secure (Seguridad) y, a continuación, elija CAs (Entidades de certificación).
3. En la lista de entidades de certificación, busque aquella para la que desea habilitar el registro automático y abra el menú de opciones mediante el icono de puntos suspensivos.
4. En el menú de opciones, elija Enable auto-registration (Habilitar registro automático).

Note

El estado de registro automático no se muestra en la lista de entidades de certificación. Para ver el estado de registro automático de una entidad de certificación, debe abrir la página Details (Detalles) de la entidad de certificación.

Configurar un certificado de entidad de certificación para admitir el registro automático (CLI)

Si ya ha registrado su certificado de entidad de certificación con AWS IoT, utilice el comando [update-ca-certificate](#) para establecer `autoRegistrationStatus` del certificado de entidad de certificación en `ENABLE`.

```
aws iot update-ca-certificate \
--certificate-id caCertificateId \
--new-auto-registration-status ENABLE
```

Si desea habilitar `autoRegistrationStatus` al registrar el certificado de entidad de certificación, utilice el 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
```

Utilice el comando [describe-ca-certificate](#) para ver el estado del certificado de entidad de certificación.

Configurar la primera conexión de un cliente para el registro automático

Cuando un cliente intenta conectarse aAWS IoTPor primera vez debe presentar un archivo que contenga el certificado de cliente firmado por el certificado de entidad de certificación como parte del protocolo de enlace TLS.

Cuando el cliente se conecta aAWS IoT, utilice el*client_certificate_filename*como archivo de certificado.AWS IoTReconoce el certificado de CA como certificado de CA registrado, registra el certificado

de cliente y establece su estado en PENDING_ACTIVATION. Esto significa que el certificado de cliente se registró automáticamente y que está a la espera de su activación. El estado del certificado de cliente debe ser ACTIVE antes de que se pueda usar para conectarse a AWS IoT.

Note

Puede aprovisionar dispositivos mediante AWS IoT Core just-in-time función de registro (JITR) sin tener que enviar toda la cadena de confianza en la primera conexión de los dispositivos a AWS IoT Core. La presentación del certificado de CA es opcional, pero el dispositivo debe enviar la [indicación de nombre de servidor \(SNI\)](#) extensión cuando se conectan.

Cuando AWS IoT registra automáticamente un certificado o cuando un cliente presenta un certificado en estado PENDING_ACTIVATION, AWS IoT publica un mensaje en el tema MQTT siguiente:

```
$aws/events/certificates/registered/caCertificateId
```

Donde *caCertificateId* es el ID del certificado de entidad de certificación que generó el certificado de cliente.

El mensaje publicado en este tema tiene la estructura siguiente:

```
{  
    "certificateId": "certificateId",  
    "caCertificateId": "caCertificateId",  
    "timestamp": timestamp,  
    "certificateStatus": "PENDING_ACTIVATION",  
    "awsAccountId": "awsAccountId",  
    "certificateRegistrationTimestamp": "certificateRegistrationTimestamp"  
}
```

Puede crear una regla que escuche este tema y realice algunas acciones. Le recomendamos crear una regla de Lambda que verifique que el certificado de cliente no se encuentre en una lista de revocación de certificados (CRL), active el certificado y cree una política y la asocie a este. La política determina a qué recursos puede tener acceso el cliente. Para obtener más información acerca de cómo crear una regla de Lambda que escuche en la \$aws/events/certificates/registered/*caCertificateId* tema y realiza estas acciones, consulte [Registro justo a tiempo de los certificados de cliente](#) en AWS IoT.

Si se produce algún error o excepción durante el registro automático de los certificados de cliente, AWS IoT envía eventos o mensajes a los inicios de sesión CloudWatch Registros. Para obtener más información acerca de cómo configurar los registros de su cuenta, consulte la [Amazon CloudWatch documentación](#).

Activar o desactivar un certificado de cliente

AWS IoT comprueba que un certificado de cliente está activo cuando autentica una conexión.

Puede crear y registrar certificados de cliente sin activarlos para que no se puedan usar hasta que deseé usarlos. También puede desactivar los certificados de cliente activos para deshabilitarlos temporalmente. Por último, puede revocar certificados de cliente para evitar que se utilicen en el futuro.

Activar un certificado de cliente (consola)

Para activar un certificado de cliente mediante la consola de AWS IoT

1. Inicie sesión en AWS La consola de administración de y abra la [AWS IoT consola](#).
2. En el panel de navegación de la izquierda, elija Secure (Seguridad) y, a continuación, elija Certificates (Certificados).
3. En la lista de certificados, busque el certificado que desea activar y abra el menú de opciones mediante el ícono de puntos suspensivos.
4. En el menú de opciones, elija Activate (Activar).

El certificado debe aparecer como Active (Activo) en la lista de certificados.

Desactivar un certificado de cliente (consola)

Para desactivar un certificado de cliente mediante la consola de AWS IoT

1. Inicie sesión enAWSLa consola de administración de y abra la[AWS IoTconsola](#).
2. En el panel de navegación de la izquierda, elija Secure (Seguridad) y, a continuación, elija Certificates (Certificados).
3. En la lista de certificados, busque el certificado que desea desactivar y abra el menú de opciones mediante el icono de puntos suspensivos.
4. En el menú de opciones, elija Deactivate (Desactivar).

El certificado debe aparecer como Inactive (Inactivo) en la lista de certificados.

Activar un certificado de cliente (CLI)

La AWS CLI proporciona el comando [update-certificate](#) para activar un certificado.

```
aws iot update-certificate \  
  --certificate-id certificateId \  
  --new-status ACTIVE
```

Si el comando se realizó correctamente, el estado del certificado será ACTIVE. Ejecute [describe-certificate](#) para ver el estado del certificado.

```
aws iot describe-certificate \  
  --certificate-id certificateId
```

Desactivar un certificado de cliente (CLI)

La AWS CLI proporciona el comando [update-certificate](#) para desactivar un certificado.

```
aws iot update-certificate \  
  --certificate-id certificateId \  
  --new-status INACTIVE
```

Si el comando se realizó correctamente, el estado del certificado será INACTIVE. Ejecute [describe-certificate](#) para ver el estado del certificado.

```
aws iot describe-certificate \  
  --certificate-id certificateId
```

Asociar un objeto o una política a un certificado de cliente

Cuando cree y registre un certificado separado de un objeto de AWS IoT, no tendrá ninguna política que autorice ninguna operación de AWS IoT, ni estará asociado con ningún objeto de AWS IoT. En esta sección se describe cómo agregar estas relaciones a un certificado registrado.

Important

Para completar estos procedimientos, debe haber creado ya el objeto o la política que desea asociar al certificado.

El certificado autentica un dispositivo conAWS IoTpara que pueda conectarse. Al adjuntar el certificado a un recurso de cosa se establece la relación entre el dispositivo (a través del certificado) y el recurso de

cosa. Para autorizar la ejecución del dispositivo AWS IoT acciones, como para permitir que el dispositivo se conecte y publique mensajes, debe adjuntarse una política adecuada al certificado del dispositivo.

Asociar un objeto a un certificado de cliente (consola)

Necesitará el nombre del objeto de cosa para completar este procedimiento.

Para asociar un objeto a un certificado registrado

1. Inicie sesión en AWS La consola de administración de y abra la [AWS IoT consola](#).
2. En el panel de navegación de la izquierda, elija Secure (Seguridad) y, a continuación, elija Certificates (Certificados).
3. En la lista de certificados, busque el certificado al que desea asociar una política, elija el icono de puntos suspensivos para abrir el menú de opciones del certificado y elija Attach thing (Asociar objeto).
4. En la ventana emergente, busque el nombre del objeto que desea asociar al certificado, elija su casilla de verificación y elija Adjuntar.

El objeto debería aparecer ahora en la lista de objetos de la página de detalles del certificado.

Asociar una política a un certificado de cliente (consola)

Necesitará el nombre del objeto de política para completar este procedimiento.

Para asociar un objeto de política a un certificado registrado

1. Inicie sesión en AWS La consola de administración de y abra la [AWS IoT consola](#).
2. En el panel de navegación de la izquierda, elija Secure (Seguridad) y, a continuación, elija Certificates (Certificados).
3. En la lista de certificados, busque el certificado al que desea asociar una política, elija el icono de puntos suspensivos para abrir el menú de opciones del certificado y elija Attach policy (Asociar política).
4. En la ventana emergente, busque el nombre de la política que desea asociar al certificado, elija su casilla de verificación y elija Adjuntar.

El objeto de política debería aparecer ahora en la lista de políticas de la página de detalles del certificado.

Asociar un objeto a un certificado de cliente (CLI)

La AWS CLI proporciona el comando [attach-thing-principal](#) para asociar un objeto a un certificado.

```
aws iot attach-thing-principal \
  --principal certificateArn \
  --thing-name thingName
```

Asociar una política a un certificado de cliente (CLI)

La AWS CLI proporciona el comando [attach-policy](#) para asociar un objeto de política a un certificado.

```
aws iot attach-policy \
  --target certificateArn \
  --policy-name policyName
```

Revocar un certificado de cliente

Si detecta actividad sospechosa en un certificado de cliente registrado, puede revocarlo para que no se pueda volver a utilizar.

Revocar un certificado de cliente (consola)

Para revocar un certificado de cliente mediante la consola de AWS IoT

1. Inicie sesión enAWSLa consola de administración de y abra la[AWS IoTconsola](#).
2. En el panel de navegación de la izquierda, elija Secure (Seguridad) y, a continuación, elija Certificates (Certificados).
3. En la lista de certificados, busque el certificado que desea revocar y abra el menú de opciones mediante el icono de puntos suspensivos.
4. En el menú de opciones, elija Revoke (Revocar).

Si el certificado se revocó correctamente, se mostrará como Revoked (Revocado) en la lista de certificados.

Revocar un certificado de cliente (CLI)

La AWS CLI proporciona el comando [update-certificate](#) para revocar un certificado.

```
aws iot update-certificate \  
  --certificate-id certificateId \  
  --new-status REVOKED
```

Si el comando se realizó correctamente, el estado del certificado será REVOKED. Ejecute [describe-certificate](#) para ver el estado del certificado.

```
aws iot describe-certificate \  
  --certificate-id certificateId
```

Transferir un certificado a otra cuenta

Certificados X.509 que pertenecen a unoCuenta de AWSse puede transferir a otroCuenta de AWS.

Para transferir un certificado X.509 desde unoCuenta de AWSa otro

1. [the section called “Iniciar una transferencia de certificados” \(p. 315\)](#)

El certificado debe desactivarse y separarse de todas las políticas y cosas antes de iniciar la transferencia.

2. [the section called “Aceptar o rechazar una transferencia de certificados” \(p. 317\)](#)

La cuenta receptora debe aceptar o rechazar explícitamente el certificado transferido. Después de que la cuenta receptora acepte el certificado, el certificado debe activarse antes de usarlo.

3. [the section called “Cancelación de una transferencia de certificados” \(p. 318\)](#)

La cuenta de origen puede cancelar una transferencia si no se ha aceptado el certificado.

Iniciar una transferencia de certificados

Puede empezar a transferir un certificado a otroCuenta de AWSmediante el uso de la[AWS IoTconsola](#)o elAWS CLI.

Iniciar una transferencia de certificados (consola)

Para completar este procedimiento, necesitará el ID del certificado que desea transferir.

Realice este procedimiento desde la cuenta con el certificado que desea transferir.

Para empezar a transferir un certificado a otraCuenta de AWS

1. Inicie sesión enAWSLa consola de administración de y abra la[AWS IoTconsola](#).
2. En el panel de navegación de la izquierda, elija Secure (Seguridad) y, a continuación, elija Certificates (Certificados).

Elija el certificado con unActivooInactivoestado que desea transferir y abra su página de detalles.

3. En el certificadoDetalles de, en elActionsmenú, si elDesactivarestá disponible, elija laDesactivaropción para desactivar el certificado.
4. En el certificadoDetalles de, en el menú de la izquierda, elijaPolíticas.
5. En el certificadoPolíticas, si hay alguna directiva adjunta al certificado, separe cada una abriendo el menú de opciones de la política y eligiendoDesacoplar.

El certificado no debe disponer de políticas adjuntas antes de continuar.

6. En el certificadoPolíticas, en el menú de la izquierda, elijaObjetos.
7. En el certificadoObjetos, si hay algo adjunto al certificado, desconecte cada uno abriendo el menú de opciones de la cosa y eligiendoDesacoplar.

El certificado no debe contener objetos adjuntos antes de continuar.

8. En el certificadoObjetos, en el menú de la izquierda, elijaDetalles de.
9. En el certificadoDetalles de, en elActionsmenú, elijalnicie transferenciapara abrirlnicie transferenciacuadro de diálogo.
10. En el navegadorlnicie transferencia, escriba elCuenta de AWSnúmero de la cuenta para recibir el certificado y un mensaje corto opcional.
11. Elegirlnicie transferenciapara transferir el certificado.

La consola debe mostrar un mensaje que indique el éxito o el error de la transferencia. Si la transferencia se inició, el estado del certificado se actualiza aTransferido.

Inicie una transferencia de certificados (CLI)

Para completar este procedimiento, necesitará la`certificateId`y la`certificateArn`del certificado que desea transferir.

Realice este procedimiento desde la cuenta con el certificado que desea transferir.

Para empezar a transferir un certificado a otroAWScuenta

1. Usar[update-certificate](#)para desactivar el certificado.

```
aws iot update-certificate --certificate-id certificateId --new-status INACTIVE
```

2. Despegue todas las políticas.

1. Usar[list-attached-policies](#)para enumerar las políticas adjuntas al certificado.

```
aws iot list-attached-policies --target certificateArn
```

2. Para cada política adjunta, utilice el[detach-policy](#)para desasociar la política.

```
aws iot detach-policy --target certificateArn --policy-name policy-name
```

3. Separa todas las cosas.

1. Usar `list-principal-things` para enumerar los elementos adjuntos al certificado.

```
aws iot list-principal-things --principal certificateArn
```

2. Para cada cosa adjunta, usa el `detach-thing-principal` para despegar la cosa.

```
aws iot detach-thing-principal --principal certificateArn --thing-name thing-name
```

4. Usar `transfer-certificate` para iniciar la transferencia de certificados.

```
aws iot transfer-certificate --certificate-id certificateId --target-aws-account account-id
```

Aceptar o rechazar una transferencia de certificados

Puede aceptar o rechazar un certificado que se le haya transferido de otra cuenta de AWS mediante el uso de la [AWS IoT consola](#) o el [AWS CLI](#).

Aceptar o rechazar una transferencia de certificados (consola)

Para completar este procedimiento, necesitará el ID del certificado que se transfirió a su cuenta.

Realice este procedimiento desde la cuenta que recibe el certificado transferido.

Para aceptar o rechazar un certificado que se ha transferido a su cuenta de AWS

1. Inicie sesión en AWS La consola de administración de y abra la [AWS IoT consola](#).
2. En el panel de navegación de la izquierda, elija Secure (Seguridad) y, a continuación, elija Certificates (Certificados).

Elija el certificado con el estado de Pending transfer que desea aceptar o rechazar y abra su página de detalles.

3. En el certificado Detalles de, en el Actions menú,
 - Para aceptar el certificado, elija Aceptación de transferencia.
 - Para no aceptar el certificado, elija Reject transfer.

Aceptar o rechazar una transferencia de certificado (CLI)

Para completar este procedimiento, necesitará la `certificateId` de la transferencia de certificado que desea aceptar o rechazar.

Realice este procedimiento desde la cuenta que recibe el certificado transferido.

Para aceptar o rechazar un certificado que se ha transferido a su cuenta de AWS

1. Usar `accept-certificate-transfer` para aceptar el certificado.

```
aws iot accept-certificate-transfer --certificate-id certificateId
```

2. Usar `reject-certificate-transfer` para rechazar el certificado.

```
aws iot reject-certificate-transfer --certificate-id certificateId
```

Cancelación de una transferencia de certificados

Puede cancelar una transferencia de certificado antes de que se acepte mediante el AWS IoT consola o el AWS CLI.

Cancelar una transferencia de certificados (consola)

Para completar este procedimiento, necesitará el ID de la transferencia de certificado que desea cancelar.

Realice este procedimiento desde la cuenta que inició la transferencia de certificados.

Para cancelar una transferencia de certificados

1. Inicie sesión en AWS La consola de administración de y abra la [AWS IoT consola](#).
 2. En el panel de navegación de la izquierda, elija Secure (Seguridad) y, a continuación, elija Certificates (Certificados).
- Elija el certificado con Transferido estado cuya transferencia desea cancelar y abra su menú de opciones.
3. En el menú de opciones del certificado, elija la Revoke Transfer opción para cancelar la transferencia de certificados.

Important

Tenga cuidado de no confundir la Revoke Transfer opción con la Revoker opción.

La Revoke Transfer cancela la transferencia de certificados, mientras que el Revoker hace que el certificado sea irreversible e inutilizable por AWS IoT.

Cancelar una transferencia de certificados (CLI)

Para completar este procedimiento, necesitará la `certificateId` de la transferencia de certificado que desea cancelar.

Realice este procedimiento desde la cuenta que inició la transferencia de certificados.

Usar `cancel-certificate-transfer` para cancelar la transferencia de certificados.

```
aws iot cancel-certificate-transfer --certificate-id certificateId
```

Usuarios, grupos y roles de IAM

Los usuarios, grupos y roles de IAM son los mecanismos estándar de administración de identidades y autenticación en AWS. Puede utilizarlos para conectarse con AWS IoT interfaces HTTP mediante el AWS SDK y AWS CLI.

Los roles de IAM también permiten AWS IoT para acceder a otros AWS recursos de la cuenta en su nombre. Por ejemplo, si desea que un dispositivo publique su estado en una tabla de DynamoDB, los roles de IAM permiten AWS IoT para interactuar con Amazon DynamoDB. Para obtener más información, consulte [Roles de IAM](#).

Para las conexiones del agente de mensajes a través de HTTP, AWS IoT autentica a los usuarios, grupos y roles de IAM mediante el proceso Signature Version 4. Para obtener información, consulte [Firma AWSSolicitudes API de](#).

Cuando se utiliza AWS Signature Version 4 con AWS IoT, los clientes deben admitir lo siguiente en su implementación de TLS:

- TLS 1.2, TLS 1.1, TLS 1.0
- Validación de la firma de certificado SHA-256 RSA

- Uno de los conjuntos de cifrado de la sección de compatibilidad del conjunto de cifrado TLS

Para obtener información, consulte [Identity and Access Management en AWS IoT \(p. 384\)](#).

Identidades de Amazon Cognito

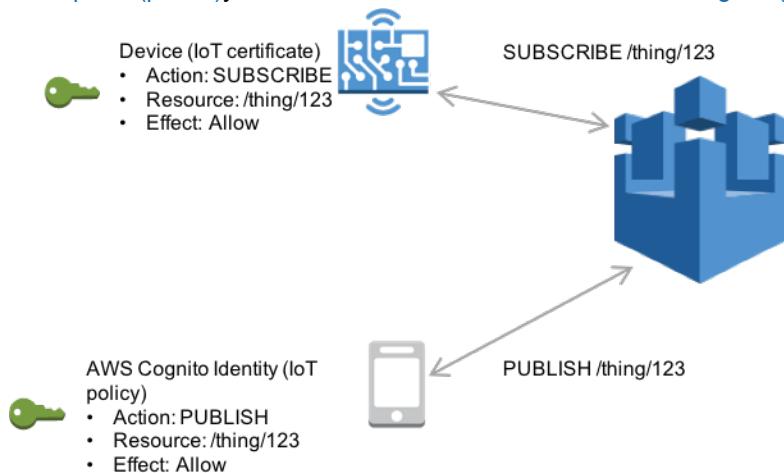
Amazon Cognito Identity le permite crear privilegios temporales y limitadosAWS credenciales para su uso en aplicaciones móviles y web. Cuando utiliza Amazon Cognito Identity, se crean grupos de identidades que crean identidades únicas para los usuarios y se autentican con proveedores de identidades como Login with Amazon, Facebook y Google. También puede utilizar identidades de Amazon Cognito con sus propias identidades autenticadas por el desarrollador. Para obtener más información, consulte [Identidad de Amazon Cognito](#).

Para utilizar Amazon Cognito Identity, se define un grupo de identidades de Amazon Cognito que está asociado a un rol de IAM. El rol de IAM está asociado a una política de IAM que concede permiso a las identidades de su grupo de identidades para acceder AWS Recursos como llamar AWS Servicios de .

Amazon Cognito Identity crea identidades no autenticadas y autenticadas. Las identidades no autenticadas se utilizan para los usuarios invitados de una aplicación móvil o web que desean usar la aplicación sin iniciar sesión. Los usuarios sin autenticar solo reciben los permisos especificados en la política de IAM asociada al grupo de identidades.

Cuando utiliza identidades autenticadas, además de la política de IAM asociada al grupo de identidades, debe asociar un AWS IoT política de Amazon Cognito Identity mediante el [AttachPolicy API](#) y otorgue permisos a un usuario individual de su AWS IoT Revisiones de. Puede utilizar el AWS IoT para asignar permisos detallados para clientes específicos y sus dispositivos.

Los usuarios autenticados y no autenticados son tipos de identidad diferentes. Si no adjunta un AWS IoT política de Amazon Cognito Identity, un usuario autenticado falla la autorización en AWS IoT y no tiene acceso a AWS IoT Recursos y acciones. Para obtener más información acerca de la creación de políticas para identidades de Amazon Cognito, consulte [Ejemplos de política de publicación/suscripción \(p. 351\)](#) y [Autorización con identidades de Amazon Cognito \(p. 373\)](#).



Autenticación personalizada

AWS IoT Core le permite definir autorizadores personalizados para que pueda administrar su propia autenticación y autorización de clientes. Esto resulta útil cuando necesita utilizar mecanismos de autenticación distintos de los que AWS IoT Core admite de forma nativa. (Para obtener más información acerca de los mecanismos de apoyo nativo, consulte [the section called “Autenticación del cliente” \(p. 297\)](#)).

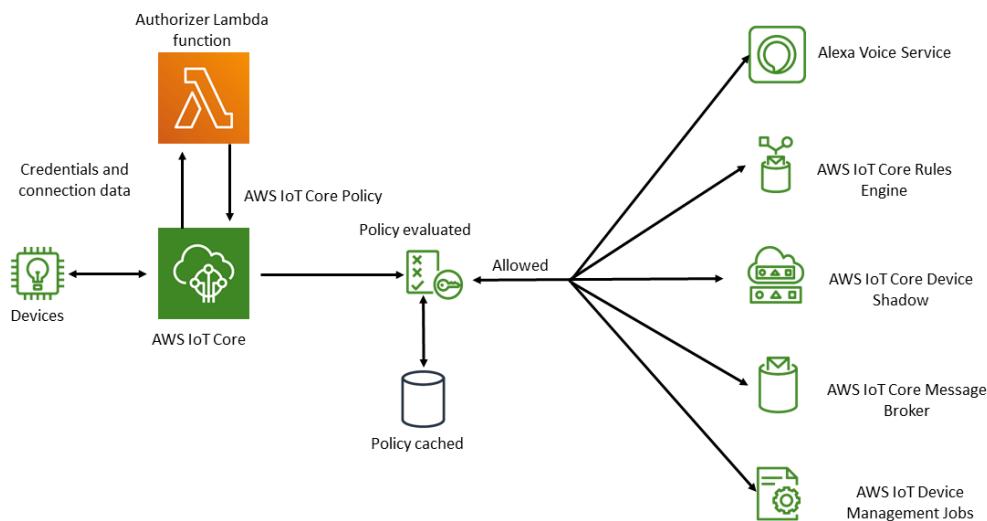
Por ejemplo, si está migrando dispositivos existentes en el campo aAWS IoT Corey estos dispositivos utilizan un token de portador personalizado o un nombre de usuario y contraseña MQTT para autenticarse, puede migrarlos aAWS IoT Coresin tener que proporcionarles nuevas identidades. Puede utilizar la autenticación personalizada con cualquiera de los protocolos de comunicación queAWS IoT Coresoportes, consulte [the section called “Protocolos de comunicación de dispositivos” \(p. 81\)](#).

Temas

- Descripción del flujo de trabajo de autenticación personalizado (p. 320)
- Creación y administración de autorizadores personalizados (p. 321)
- Conexión aAWS IoT Coremediante autenticación personalizada (p. 327)
- Solución de problemas de los autorizadores (p. 328)

Descripción del flujo de trabajo de autenticación personalizado

La autenticación personalizada le permite definir cómo autenticar y autorizar clientes mediante [recursos de autorizadores](#). Cada autorizador contiene una referencia a una función Lambda administrada por el cliente, una clave pública opcional para validar las credenciales de dispositivo e información de configuración adicional. El siguiente diagrama ilustra el flujo de trabajo de autorización para la autenticación personalizada enAWS IoT Core.



AWS IoT Core flujo de trabajo de autenticación y autorización personalizado

En la lista siguiente se explica cada paso del flujo de trabajo de autenticación y autorización personalizado.

1. Un dispositivo se conecta a la de un clienteAWS IoT Core punto de enlace de datos mediante una de las admitidas [the section called “Protocolos de comunicación de dispositivos” \(p. 81\)](#). El dispositivo pasa credenciales en los campos de encabezado de la solicitud o en los parámetros de consulta (para HTTP Publish o MQTT a través de WebSockets protocolos) o en el campo nombre de usuario y contraseña del mensaje MQTT CONNECT (para los protocolos MQTT y MQTT sobre WebSockets).
2. AWS IoT Core comprueba una de las dos condiciones:
 - La solicitud entrante especifica un autorizador.

- La AWS IoT Core es el punto final de datos que recibe la solicitud tiene configurado un autorizador predeterminado para ella.

Si AWS IoT Core encuentra un autorizador de cualquiera de estas formas, AWS IoT Core activa la función Lambda asociada al autorizador.

3. (Opcional) Si has habilitado la firma de tokens, AWS IoT Core valida la firma de la solicitud utilizando la clave pública almacenada en el autorizador antes de activar la función Lambda. Si se produce un error en la validación, AWS IoT Core detiene la solicitud sin invocar la función Lambda.
4. La función Lambda recibe las credenciales y los metadatos de conexión de la solicitud y toma una decisión de autenticación.
5. La función Lambda devuelve los resultados de la decisión de autenticación y AWS IoT Core documenta la política que especifica qué acciones se permiten en la conexión. La función Lambda también devuelve información que especifica con qué frecuencia AWS IoT Core valida las credenciales de la solicitud mediante la invocación de la función Lambda.
6. AWS IoT Core evalúa la actividad de la conexión con la política que ha recibido de la función Lambda.

Consideraciones de escalado

Dado que una función Lambda gestiona la autenticación y la autorización de su autorizador, la función está sujeta a los límites de precios y servicios de Lambda, como la tasa de ejecución simultánea. Para obtener más información acerca de los precios de Lambda, consulte [Precios de Lambda](#). Puede administrar la carga de la función Lambda ajustando `lambda.refreshAfterInSeconds` y `lambda.disconnectAfterInSeconds` parámetros de la respuesta de la función Lambda. Para obtener más información acerca del contenido de su respuesta de función de Lambda, consulte [the section called “Definición de la función Lambda” \(p. 322\)](#).

Note

Si deja habilitada la firma, puede evitar la activación excesiva de su Lambda por parte de clientes no reconocidos. Tenga en cuenta esto antes de deshabilitar el inicio de sesión en su autorizador.

Creación y administración de autorizadores personalizados

AWS IoT Core implementa esquemas de autenticación y autorización personalizados mediante [recursos de autorizadores](#). Cada autorizador consta de los siguientes componentes:

- Nombre: Una única cadena definida por el usuario que identifica al autorizador.
- ARN de la función Lambda: El Nombre de recurso de Amazon (ARN) de la función Lambda que implementa la lógica de autorización y autenticación.
- Nombre de clave de token: Nombre clave utilizado para extraer el token de los encabezados HTTP, los parámetros de consulta o el nombre de usuario de MQTT CONNECT para llevar a cabo la validación de la firma. Este valor es obligatorio si la firma de esté habilitada en su autorizador.
- Indicador de firma deshabilitado (opcional): Un valor booleano que especifica si se debe deshabilitar el requisito de firma de las credenciales. Esto resulta útil para escenarios en los que la firma de las credenciales no tiene sentido, como esquemas de autenticación que utilizan el nombre de usuario y la contraseña de MQTT. El valor predeterminado es `false`, por lo que la firma está habilitada de forma predeterminada.
- Clave pública de firma de tokens: La clave pública de que AWS IoT Core utiliza para validar la firma del token. Su longitud mínima es 2.048 bits. Este valor es obligatorio si la firma de esté habilitada en su autorizador.

Lambda le cobra por el número de veces que se ejecuta la función Lambda y por el tiempo que tarda en ejecutarse el código de su función. Para obtener más información acerca de los precios de Lambda, consulte [Precios de Lambda](#). Para obtener más información acerca de la creación de funciones de Lambda, consulte la [Guía para desarrolladores de Lambda](#).

Note

Si deja habilitada la firma, puede evitar la activación excesiva de su Lambda por parte de clientes no reconocidos. Tenga en cuenta esto antes de deshabilitar el inicio de sesión en su autorizador.

Definición de la función Lambda

Cuando AWS IoT Core invoca al autorizador, activa el Lambda asociado al autorizador con un evento que contiene el siguiente objeto JSON. El objeto JSON de ejemplo contiene todos los campos posibles. Los campos que no sean relevantes para la solicitud de conexión no se incluyen.

```
{  
    "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.  
    },  
}
```

La función Lambda debe utilizar esta información para autenticar la conexión entrante y decidir qué acciones se permiten en la conexión. La función debe enviar una respuesta que contenga los siguientes valores.

- **isAuthenticated**: un valor booleano que indica si la solicitud está autenticada.
- **principalId**: una cadena alfanumérica que actúa como identificador del token enviado por la solicitud de autorización personalizada. El valor debe ser una cadena alfanumérica con al menos uno y no más de 128 caracteres y coincidir con este patrón de expresión regular (regex):([a-zA-Z0-9]{1,128}).
- **policyDocuments**: Una lista de formato JSONAWS IoT Core documentos de políticas Para obtener más información acerca de la creación de AWS IoT Core políticas de, consulte [the section called “Políticas de AWS IoT Core” \(p. 333\)](#). El número máximo de documentos de póliza es de 10 documentos de política. Cada documento de política puede contener un máximo de 2.048 caracteres.
- **disconnectAfterInSeconds**: un entero que especifica la duración máxima (en segundos) de la conexión al AWS IoT Core gateway de. El valor mínimo es de 300 segundos y el máximo es de 86 400 segundos.
- **refreshAfterInSeconds**: un entero que especifica el intervalo entre las actualizaciones de la política de. Cuando transcurra este intervalo, AWS IoT Core invoca la función Lambda para permitir actualizaciones de políticas. El valor mínimo es de 300 segundos y el máximo es de 86 400 segundos.

El siguiente objeto JSON contiene un ejemplo de respuesta que puede enviar la función Lambda.

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

La `policyDocument` valor debe contener un valor válido AWS IoT Core documento de política. Para obtener más información acerca de AWS IoT Core políticas de, consulte [the section called “Políticas de AWS IoT Core” \(p. 333\)](#). En MQTT sobre TLS y MQTT a través de conexiones WebSockets, AWS IoT Core almacena en caché esta política durante el intervalo especificado en el valor `refreshAfterInSeconds`. En el caso de las conexiones HTTP, se llama a la función Lambda para cada solicitud de autorización, a menos que su dispositivo utilice conexiones persistentes HTTP (también denominadas HTTP keep-alive o reutilización de conexión HTTP), puede elegir habilitar el almacenamiento en caché al configurar el autorizador. Durante este intervalo, AWS IoT Core autoriza acciones en una conexión establecida contra esta política almacenada en caché sin volver a activar la función Lambda. Si se producen errores durante la autenticación personalizada, AWS IoT Core finaliza la conexión. AWS IoT Core también finaliza la conexión si ha estado abierta durante más tiempo que el valor especificado en el `disconnectAfterInSeconds` parámetro.

Los siguientes ejemplos de JavaScript contiene un ejemplo de la función Lambda Node.js que busca una contraseña en el mensaje MQTT Connect con un valor `test` y devuelve una política que concede permiso para conectarse a AWS IoT Core con un cliente llamado `myClientName` y publica en un tema que contenga el mismo nombre de cliente. Si no encuentra la contraseña esperada, devuelve una política que niega esas dos acciones.

```
// 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;
}

```

La función Lambda anterior devuelve el siguiente JSON cuando recibe la contraseña esperada de `testen` en el mensaje MQTT Connect. Los valores de `password` y `principalId` properties serán los valores del mensaje 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
    }
}
```

Creación de un autorizador

Puede crear un autorizador a través de la[CreateAuthorizer API](#). En el siguiente ejemplo se describe el comando de.

```
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]
```

Puede utilizar el `signing-disabled` para excluirse de la validación de firma para cada invocación de su autorizador. Recomendamos encarecidamente que no deshabilite la firma a menos que tenga que hacerlo. La validación de firmas lo protege contra invocaciones excesivas de la función Lambda desde dispositivos desconocidos. No se puede actualizar el `signing-disabled` de un autorizador después de crearlo. Para cambiar este comportamiento, debe crear otro autorizador personalizado con un valor de diferente `signing-disabled` parámetro.

Valores de `tokenKeyName` y `tokenSigningPublicKeys` los parámetros son opcionales si ha deshabilitado la firma. Son valores obligatorios si la firma está habilitada.

Después de crear la función Lambda y el autorizador personalizado, debe conceder explícitamente el AWS IoT Core permiso de servicio para invocar la función en su nombre. Puede hacer esto con el siguiente comando de.

```
aws lambda add-permission --function-name <lambda_function_name> --principal
iot.amazonaws.com --source-arn <authorizer_arn> --statement-id Id-123 --action
"lambda:InvokeFunction"
```

Comprobación de los autorizadores

Puede utilizar el[TestInvokeAuthorizer API](#) para probar los valores de invocación y devolución de su autorizador. Esta API le permite especificar metadatos de protocolo y probar la validación de firma en su autorizador.

En las siguientes pestañas se muestra cómo utilizar la AWS CLI para probar su 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
```

El valor del `--token-signature` parámetro es el token firmado. Para obtener información sobre cómo obtener este valor, consulte [the section called “Firmar el token” \(p. 328\)](#).

Si el autorizador de toma un nombre de usuario y contraseña, puede transferir esta información utilizando el `--mqtt-context` parámetro. En las siguientes pestañas se muestra cómo utilizar la `TestInvokeAuthorizer` API para enviar un objeto JSON que contiene un nombre de usuario, contraseña y nombre de cliente a su 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"}'
```

La contraseña debe estar codificada con base64. El siguiente ejemplo muestra cómo codificar una contraseña en un entorno similar a Unix.

```
echo -n PASSWORD | base64
```

Administración de autorizadores personalizados

Puede administrar sus autorizadores mediante las siguientes API.

- [ListAuthorizers](#): Muestra todos los autorizadores de la cuenta.
- [DescribeAuthorizer](#): Muestra las propiedades del autorizador especificado. Estos valores incluyen fecha de creación, fecha de última modificación y otros atributos.
- [SetDefaultAuthorizer](#): Especifica el autorizador predeterminado para la AWS IoT Core puntos finales de datos. AWS IoT Core utiliza este autorizador si un dispositivo no pasa AWS IoT Core credenciales y no especifica un autorizador. Para obtener más información sobre el uso de AWS IoT Core credenciales, consulte [the section called “Autenticación del cliente” \(p. 297\)](#).
- [UpdateAuthorizer](#): Cambia el estado, el nombre de la clave de token o las claves públicas del autorizador especificado.
- [DeleteAuthorizer](#): Elimina el autorizador especificado.

Note

No puedes actualizar el requisito de firma de un autorizador. Esto significa que no puedes deshabilitar el inicio de sesión en un autorizador existente que lo requiera. Tampoco puedes requerir que inicies sesión en un autorizador existente que no lo requiere.

Conexión aAWS IoT Coremediante autenticación personalizada

Los dispositivos se pueden conectar aAWS IoT Coremediante la autenticación personalizada con cualquier protocolo que AWS IoT Coreadmita mensajería de dispositivos. Para obtener más información acerca de los protocolos de comunicación admitidos, consulte [the section called “Protocolos de comunicación de dispositivos” \(p. 81\)](#). Los datos de conexión que transfiere a la función Lambda autorizador dependen del protocolo que utilice. Para obtener más información acerca de cómo crear su función de Lambda de autorizador, consulte [the section called “Definición de la función Lambda” \(p. 322\)](#). En las secciones siguientes se explica cómo conectarse a la autenticación mediante cada protocolo compatible.

HTTP

Dispositivos que envían datos aAWS IoT Coremediante el uso de la [API de publicación HTTP](#) puede pasar credenciales a través de encabezados de solicitud o parámetros de consulta en sus solicitudes HTTP POST. Los dispositivos pueden especificar un autorizador para invocar mediante el `x-amz-customauthorizer-name`encabezado o parámetro de consulta. Si tienes habilitada la firma de tokens en tu autorizador, debes pasar la `token-key-name`y `x-amz-customauthorizer-signature`en encabezados de solicitud o parámetros de consulta. Tenga en cuenta que la `token-signature`el valor debe estar codificado en URL cuando se utiliza JavaScript desde dentro del navegador.

En las siguientes solicitudes de ejemplo se muestra cómo pasa estos parámetros tanto en los encabezados de solicitud como en los parámetros 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 conectados aAWS IoT Coremediante una conexión MQTT puede pasar credenciales a través de la `username`y `password`campos de mensajes MQTT. La `username`value también puede contener opcionalmente una cadena de consulta que pase valores adicionales (incluido un token, una firma y un nombre de autorizador) al autorizador. Puede utilizar esta cadena de consulta si desea utilizar un esquema de autenticación basado en tokens en lugar de `username`y `password`valores.

Note

Los datos del campo de contraseña están codificados en base64AWS IoT Core. Tu función Lambda debe decodificarlo.

En el siguiente ejemplo se incluye un `username`cadena que contiene parámetros adicionales que especifican un token y una firma.

```
username?x-amz-customauthorizer-name=authorizer-name&x-amz-customauthorizer-
signature=token-signature&token-key-name=token-value
```

Para invocar a un autorizador, los dispositivos que se conectan aAWS IoT Coremediante MQTT y la autenticación personalizada deben conectarse en el puerto 443. También deben pasar la extensión TLS de negociación de protocolo de capa de aplicación (ALPN) con un valor de `mqqt` la extensión Indicación de nombre de servidor (SNI) con el nombre de host de suAWS IoT Corepuerto final de datos. Para obtener más información acerca de estos valores de la tarea, consulte [the section called “Protocolos de](#)

comunicación de dispositivos” (p. 81). El V2AWS IoTSDK de dispositivos, SDK móviles yAWS IoTClientede dispositivos (p. 1274)puede configurar ambas extensiones.

MQTT sobre WebSockets

Dispositivos conectados aAWS IoT Coreutilizando MQTT over WebSockets puede transmitir credenciales de una de las dos siguientes formas.

- Mediante encabezados de solicitud o parámetros de consulta en la solicitud HTTP UPGRADE para establecer el WebSockets conexión de.
- A través delusernamey passwordcampos del mensaje MQTT CONNECT.

Si pasa credenciales a través del mensaje MQTT connect, se requieren las extensiones TLS ALPN y SNI. Para obtener más información acerca de estas extensiones, consulte[the section called “MQTT” \(p. 327\)](#). El siguiente ejemplo muestra cómo transferir las credenciales a través de la solicitud de actualización de 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
```

Firmar el token

Debe firmar el token con la clave privada del key pair pública/privada que utilizó encreate-authorizerLlame a. En los ejemplos siguientes se muestra cómo crear la firma de token mediante un comando similar a UNIX y JavaScript. Utilizan el algoritmo hash SHA-256 para codificar la firma.

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')
```

Solución de problemas de los autorizadores

En este tema se describen los problemas comunes que pueden causar problemas en los flujos de trabajo de autenticación personalizados y los pasos para resolverlos. Para solucionar problemas

de forma más eficaz, active CloudWatch Registros de paraAWS IoT Core establezca el nivel de registro enDEPURAR. Puedes habilitar CloudWatch inicios de sesión en elAWS IoT CoreConsola de (<https://console.aws.amazon.com/iot>). Para obtener más información acerca de la habilitación y configuración de registros paraAWS IoT Core, consulte [the section called “Configuración de registros de AWS IoT” \(p. 427\)](#).

Note

Si deja el nivel de registro enDEPURAR durante largos períodos de tiempo, CloudWatch podría almacenar grandes cantidades de datos de registro. Esto puede aumentar los cargos de CloudWatch. Considere utilizar el registro basado en recursos para aumentar la verbosidad de solo los dispositivos de un grupo de cosas concreto. Para obtener más información acerca de los registros basados en recursos, consulte [the section called “Configuración de registros de AWS IoT” \(p. 427\)](#). Además, cuando haya terminado de solucionar problemas, reduzca el nivel de registro a un nivel menos detallado.

Antes de comenzar a solucionar problemas, revise [the section called “Descripción del flujo de trabajo de autenticación personalizado” \(p. 320\)](#) Para obtener una perspectiva general del proceso de autenticación personalizado. Esto le ayuda a comprender dónde buscar el origen de un problema.

En este tema se analizan las dos áreas siguientes que debe investigar.

- Problemas relacionados con la función Lambda de su autorizador.
- Problemas relacionados con tu dispositivo.

Comprueba si hay problemas en la función Lambda de tu autorizador

Lleve a cabo los siguientes pasos para asegurarse de que los intentos de conexión de los dispositivos invoquen la función Lambda.

1. Verifique qué función Lambda está asociada a su autorizador.

Puede hacerlo llamando al [DescribeAuthorizer](#) API o haciendo clic en el autorizador deseado en el [Seguridad](#) Sección sobre de la AWS IoT Coreconsola de .

2. Compruebe las métricas de invocación de la función Lambda. Realice los siguientes pasos para hacerlo.
 - a. Abra el icono [Lambda](#) Consola de (<https://console.aws.amazon.com/lambda/>) y seleccione la función que está asociada con su autorizador.
 - b. Elija el icono [Monitoreo](#) y visualización de métricas para el período de tiempo que es relevante para tu problema.
3. Si no ves invocaciones, verifica que AWS IoT Core tiene permiso para invocar su función de Lambda. Si ve invocaciones, vaya al siguiente paso. Lleve a cabo los siguientes pasos para comprobar que la función Lambda tiene los permisos necesarios.
 - a. Elija el icono [Permisos](#) pestaña para tu función en el AWS Lambda consola de .
 - b. Busque el [Política](#) basada en recursos en la parte inferior de la página. Si la función Lambda tiene los permisos necesarios, la política se parece al siguiente ejemplo.

```
{  
    "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 a `InvokeFunction` permiso de su función para el AWS IoT Core director. Si no lo ve, tendrá que añadirlo utilizando el [AddPermission API](#). En el siguiente ejemplo se muestra cómo hacerlo mediante la AWS CLI.

```
aws lambda add-permission --function-name FunctionName --principal
iot.amazonaws.com --source-arn AuthorizerARN --statement-id Id-123 --action
"lambda:InvokeFunction"
```

4. Si ves invocaciones, verifica que no haya errores. Un error podría indicar que la función Lambda no gestiona correctamente el evento de conexión que AWS IoT Core lo envía.

Para obtener información sobre cómo gestionar el evento en la función Lambda, consulte [the section called “Definición de la función Lambda” \(p. 322\)](#). Puede utilizar la función de prueba en la AWS Lambda Consola de (<https://console.aws.amazon.com/lambda/>) a los valores de prueba de código duro de la función para asegurarse de que la función gestiona correctamente los eventos.

5. Si ves invocaciones sin errores, pero tus dispositivos no pueden conectarse (ni publicar, suscribirse y recibir mensajes), el problema podría ser que la política que devuelve la función Lambda no otorgue permisos para las acciones que intentan realizar tus dispositivos. Lleve a cabo los siguientes pasos para determinar si hay algo incorrecto en la política que devuelve la función.
- Usar Amazon CloudWatch Consulta de Logs Insights para escanear los registros durante un breve período de tiempo para comprobar si hay errores. En la siguiente consulta de ejemplo se ordenan los eventos por marca de hora y se buscan errores.

```
display clientId, eventType, status, @timestamp | sort @timestamp desc | filter
status = "Failure"
```

- Actualiza la función Lambda para registrar los datos a los que vuelve AWS IoT Core el evento que activa la función. Puede utilizar estos registros para inspeccionar la política que crea la función.

Investigación de problemas de dispositivo

Si no encuentra ningún problema al invocar la función Lambda o con la política que devuelve la función, busque problemas con los intentos de conexión de los dispositivos. Las solicitudes de conexión mal formadas pueden provocar que AWS IoT Core no active su autorizador. Pueden producirse problemas de conexión tanto en las capas TLS como en las aplicaciones.

Posibles problemas de capa TLS:

- Los clientes deben pasar un encabezado de nombre de host (HTTP, MQTT a través de WebSockets) o la extensión TLS de indicación de nombre del servidor (HTTP, MQTT sobre WebSockets, MQTT) en

todas las solicitudes de autenticación personalizadas. En ambos casos, el valor pasado debe coincidir con uno de los de tu cuentaAWS IoT Corepuntos finales de datos. Estos son los extremos que se devuelven al ejecutar los siguientes comandos de la CLI.

- `aws iot describe-endpoint --endpoint-type iot:data-ats`
- `aws iot describe-endpoint --endpoint-type(heredado) VeriSign Puntos de enlace)`
- Los dispositivos que utilizan autenticación personalizada para conexiones MQTT también deben pasar la extensión TLS de negociación de protocolo de capa de aplicación (ALPN) con un valor de `mqqt`.
- La autenticación personalizada está disponible actualmente solo en el puerto 443.

Posibles problemas de capa de aplicaciones:

- Si la firma está habilitada (`signingDisabled` falso en su autorizador), busque los siguientes problemas de firma.
 - Asegúrese de pasar la firma del token en `amz-customauthorizer-signature` encabezado o en un parámetro de cadena de consulta.
 - Asegúrese de que el servicio no esté firmando ningún valor que no sea el token.
 - Asegúrese de pasar el token en el encabezado o parámetro de consulta que especificó en el `token-key-name` en tu autorizador.
- Asegúrese de que el nombre del autorizador que pasa en `amz-customauthorizer-name` el parámetro de cadena de consulta o encabezado es válido o tiene un autorizador predeterminado definido para su cuenta.

Autorización

Autorización es el proceso de concesión de permisos a una identidad autenticada. Concede permisos en AWS IoT Core con AWS IoT Core políticas de IAM. En este tema se abordan las políticas de AWS IoT Core. Para obtener más información acerca de las políticas de IAM, consulte [Identity and Access Management en AWS IoT \(p. 384\)](#) y [Políticas administradas de IAM \(p. 406\)](#).

Las políticas de AWS IoT Core determinan lo que puede hacer una identidad autenticada. Los dispositivos y las aplicaciones móviles, web y de escritorio utilizan identidades autenticadas. Una identidad autenticada puede ser incluso un usuario que ejecute comandos de la CLI de AWS IoT Core. Una identidad puede ejecutar operaciones de AWS IoT Core solo si cuenta con una política que le conceda permiso para dichas operaciones.

Ambos AWS IoT Core políticas y políticas de IAM se utilizan con AWS IoT Core para controlar las operaciones de una identidad (también denominada principal) que puede funcionar. El tipo de política que utilice depende del tipo de identidad que esté utilizando para autenticarse en AWS IoT Core.

Las operaciones de AWS IoT Core se dividen en dos grupos:

- La API del plano de control le permite realizar tareas administrativas, como crear o actualizar certificados, objetos, reglas, etc.
- La API del plano de datos le permite enviar datos a AWS IoT Core y recibir datos de este servicio.

El tipo de política que utilice depende de si utiliza la API del plano de control o la del plano de datos.

En la tabla siguiente se muestran los tipos de identidad, los protocolos que utilizan y los tipos de políticas que se pueden utilizar para la autorización.

API del plano de datos de AWS IoT Core y tipos de políticas

Protocolo y mecanismo de autenticación	SDK	Tipo de identidad	Tipo de política		
MQTT sobre TLS/TCP, autenticación mutua TLS (puerto 8883 o 443) [†] (p. 82)	SDK de dispositivos de AWS IoT	Certificados X.509	Política de AWS IoT Core		
MQTT sobre HTTPS/WebSocket, AWS Autenticación Sigv4 (puerto 443)	AWSMobile SDK	Identidad de Amazon Cognito autenticada	IAM y AWS IoT Core Políticas de		
		Identidad de Amazon Cognito no autenticada	Política de IAM		
		IAM, o identidad federada	Política de IAM		
HTTPS, AWS Autenticación de firma Versión 4 (puerto 443)	AWS CLI	Amazon Cognito, IAM o identidad federada	Política de IAM		
HTTPS, autenticación mutua TLS (puerto 8443)	Sin compatibilidad de SDK	Certificados X.509	Política de AWS IoT Core		
HTTPS sobre autenticación personalizada (Puerto 443)	SDK de dispositivos de AWS IoT	Autorizador personalizado	Política de autorizador personalizada		

API del plano de control y tipos de política de AWS IoT Core

Protocolo y mecanismo de autenticación	SDK	Tipo de identidad	Tipo de política		
HTTPSAWS Autenticación de firma Versión 4 (puerto 443)	AWS CLI	Identidad Amazon Cognito	Política de IAM		
		IAM, o identidad federada	Política de IAM		

AWS IoT Core las políticas de están asociadas a certificados X.509 o identidades de Amazon Cognito. Las políticas de IAM se asocian a un usuario, grupo o rol de IAM. Si utiliza el AWS IoT Consola de o AWS IoT Core CLI para asociar la política (a un certificado o Amazon Cognito Identity), utilice una AWS IoT Core política. De lo contrario, utilice una política de IAM.

Las autorizaciones basadas en políticas son una herramienta muy eficaz. Le dan un control completo sobre lo que un dispositivo, usuario o aplicación puede hacer en AWS IoT Core. Por ejemplo, tomemos el caso de un dispositivo que se conecte con AWS IoT Core mediante un certificado. Puede permitir que el dispositivo tenga acceso a todos los temas MQTT, o bien puede restringir su acceso a un único tema. O tomemos, por ejemplo, el caso de un usuario que ejecute comandos de la CLI en una línea de comandos. Si aplica una política, puede permitirle o denegarle el acceso a cualquier comando o recurso de AWS IoT Core. También puede controlar el acceso de una aplicación a los recursos de AWS IoT Core.

Los cambios realizados en una política pueden tardar unos minutos en hacerse efectivos debido a cómo AWS IoT mantiene en caché los documentos de política. Es decir, puede tardar unos minutos en acceder a un recurso al que se le ha concedido acceso recientemente y un recurso puede estar disponible durante varios minutos después de que se haya revocado su acceso.

AWS Training and Certification

Para obtener información sobre la autorización en AWS IoT Core, tome el [Sumérgete en AWS IoT Core Autenticación y autorización](#) curso sobre el AWS Sitio web de formación y certificación.

Políticas de AWS IoT Core

Las políticas de AWS IoT Core son documentos JSON. Siguen las mismas convenciones que las políticas de IAM. AWS IoT Core admite políticas con un nombre que permita que muchas identidades puedan hacer referencia al mismo documento de política. Las políticas con nombre cuentan con varias versiones para facilitar su restauración.

Las políticas de AWS IoT Core le permiten controlar el acceso al plano de datos de AWS IoT Core. La AWS IoT Core plano de datos consiste en operaciones que le permiten conectarse al AWS IoT Core agente de mensajes de, envíe y reciba mensajes MQTT y obtenga o actualice Device Shadow de una cosa.

Una política de AWS IoT Core es un documento JSON que contiene una o varias declaraciones de política. Cada instrucción contiene:

- **Effect**, que especifica si se permite o se deniega la acción.
- **Action**, que especifica la acción que la política permite o deniega.
- **Resource**, que especifica los recursos en los que se permite o se deniega la acción.

Los cambios realizados en una política pueden tardar unos minutos en hacerse efectivos debido a cómo AWS IoT mantiene en caché los documentos de política. Es decir, puede tardar unos minutos en acceder a un recurso al que se le ha concedido acceso recientemente y un recurso puede estar disponible durante varios minutos después de que se haya revocado su acceso.

Temas

- [Acciones de política de AWS IoT Core \(p. 334\)](#)
- [Recursos de acción de AWS IoT Core \(p. 336\)](#)
- [Variables de las políticas de AWS IoT Core \(p. 337\)](#)
- [Prevención del suplente confuso entre servicios \(p. 343\)](#)
- [Ejemplos de políticas de AWS IoT Core \(p. 344\)](#)
- [Autorización con identidades de Amazon Cognito \(p. 373\)](#)

Acciones de política de AWS IoT Core

AWS IoT Core define las siguientes acciones de política:

Acciones de política de MQTT

`iot:Connect`

Representa el permiso para conectarse con el agente de mensajes de AWS IoT Core. El permiso `iot:Connect` se comprueba cada vez que se envía una solicitud CONNECT al agente. El agente de mensajes no permite que haya dos clientes con el mismo ID de cliente conectados simultáneamente. Despues de que el segundo cliente se conecta, el agente cierra la conexión existente. Usar `iot:Connect` para garantizar que solo puedan conectarse los clientes autorizados que utilizan un ID de cliente específico.

`iot:GetRetainedMessage`

Representa el permiso para obtener el contenido de un solo mensaje retenido. Los mensajes retenidos son los mensajes que se publicaron con la marca RETAIN establecida y almacenada por AWS IoT Core. Para obtener permiso para obtener una lista de todos los mensajes de la cuenta, consulte [iot>ListRetainedMessages \(p. 334\)](#).

`iot>ListRetainedMessages`

Representa el permiso para recuperar información de resumen sobre los mensajes retenidos de la cuenta, pero no sobre el contenido de los mensajes. Los mensajes retenidos son los mensajes que se publicaron con la marca RETAIN establecida y almacenada por AWS IoT Core. El ARN de recursos especificado para esta acción debe ser*. Para obtener permiso para obtener el contenido de un único mensaje conservado, consulte [iot:GetRetainedMessage \(p. 334\)](#).

`iot:Publish`

Representa el permiso para publicar un tema MQTT. Este permiso se comprueba cada vez que se envía una solicitud PUBLISH al agente. Puede utilizar esta opción para permitir a los clientes publicar en determinados patrones de tema.

Note

Para conceder el permiso `iot:Publish`, también debe otorgar el permiso `iot:Connect`.

`iot:Receive`

Representa el permiso para recibir un mensaje de AWS IoT Core. La `iot:Receive` el permiso se confirma cada vez que se entrega un mensaje a un cliente. Dado que este permiso se comprueba en cada entrega, puede utilizarlo para revocar permisos a clientes que están suscritos en ese momento a un tema.

`iot:RetainPublish`

Representa el permiso para publicar un mensaje de MQTT con el conjunto de indicadores RETAIN.

Note

Para conceder el permiso `iot:RetainPublish`, también debe otorgar el permiso `iot:Publish`.

`iot:Subscribe`

Representa el permiso para suscribirse a un filtro de temas. Este permiso se comprueba cada vez que se envía una solicitud SUBSCRIBE al agente. Se utiliza para permitir a los clientes suscribirse a temas que coinciden con patrones de tema específicos.

Note

Para conceder el permiso `iot:Subscribe`, también debe otorgar el permiso `iot:Connect`.

Acciones de política de sombra de dispositivo

`iot:DeleteThingShadow`

Representa el permiso para eliminar la sombra de dispositivo de una objeto.

Laiot:`DeleteThingShadow`El permiso se comprueba cada vez que se presenta una solicitud para eliminar el contenido Device Shadow de una objeto.

`iot:GetThingShadow`

Representa el permiso para recuperar la sombra de dispositivo de una objeto.

Laiot:`GetThingShadow`El permiso se comprueba cada vez que se presenta una solicitud para recuperar el contenido Device Shadow de una cosa.

`iot>ListNamedShadowsForThing`

Representa el permiso para enumerar una cosa llamada Sombras.

Laiot:`ListNamedShadowsForThing`El permiso se comprueba cada vez que se presenta una solicitud para obtener una lista de una cosa llamada Sombras.

`iot:UpdateThingShadow`

Representa el permiso para actualizar la sombra de un dispositivo. Laiot:`UpdateThingShadow`El permiso se comprueba cada vez que se presenta una solicitud para actualizar el contenido Device Shadow de una cosa.

Note

Las acciones de política de ejecución de trabajo se aplican únicamente al punto de enlace HTTP TLS. Si utiliza el punto de enlace MQTT, debe utilizar las acciones de política MQTT definidas en este tema.

Para ver un ejemplo de una política de ejecución de trabajos que lo demuestra, consulte [the section called “Ejemplo de política de trabajo básica” \(p. 372\)](#) que funciona con el protocolo de MQTT.

Acciones de política de AWS IoT Core de ejecución de trabajo

`iot:DescribeJobExecution`

Representa el permiso para recuperar una ejecución de trabajo para un objeto determinado. El permiso `iot:DescribeJobExecution` se comprueba cada vez que se presenta una solicitud para obtener la ejecución de un trabajo.

`iot:GetPendingJobExecutions`

Representa el permiso para recuperar la lista de trabajos que no están en un estado final para un objeto. El permiso `iot:GetPendingJobExecutions` se comprueba cada vez que se presenta una solicitud para recuperar la lista.

`iot:UpdateJobExecution`

Representa el permiso para actualizar una ejecución de trabajo. El permiso `iot:UpdateJobExecution` se comprueba cada vez que se presenta una solicitud para actualizar el estado de una ejecución de trabajo.

`iot:StartNextPendingJobExecution`

Representa el permiso para obtener e iniciar la próxima ejecución de trabajo pendiente para un objeto, es decir, para actualizar una ejecución de trabajo con un estado QUEUED a IN_PROGRESS. El permiso `iot:StartNextPendingJobExecution` se comprueba cada vez que se presenta una solicitud para iniciar la siguiente ejecución de trabajo pendiente.

AWS IoT CoreAcción de política de proveedor de credenciales

`iot:AssumeRoleWithCertificate`

Representa el permiso para llamarAWS IoT Coreproveedor de credenciales para asumir un rol de IAM con autenticación basada en certificados. La*iot:AssumeRoleWithCertificate*El permiso se comprueba cada vez que se presenta una solicitud aAWS IoT Coreproveedor de credenciales para asumir un rol.

Recursos de acción de AWS IoT Core

Para especificar un recurso para una acción de política de AWS IoT Core, debe utilizar el ARN del recurso. Todos los ARN de recursos tienen la forma siguiente:

`arn:aws:iot:<region>:<AWS-account-ID>:<Resource-type>/<Resource-name>`

En la tabla siguiente se muestra el recurso que debe especificarse para cada tipo de acción:

Acción	Tipo de recurso	Nombre del recurso	Ejemplo de ARN
<code>iot:Connect</code>	<code>client</code>	El ID de cliente del cliente	<code>arn:aws:iot:us-east-1:123456789012:client/myClientId</code>
<code>iot:DeleteThingShadow</code>	<code>thing</code>	El nombre del objeto	<code>arn:aws:iot:us-east-1:123456789012:thing/thingOne</code>
<code>iot:DescribeJobExecution</code>	<code>thing</code>	El nombre del objeto	<code>arn:aws:iot:us-east-1:123456789012:thing/thingOne`</code>
<code>iot:GetPendingJobExecutions</code>	<code>thing</code>	El nombre del objeto	<code>arn:aws:iot:us-east-1:123456789012:thing/thingOne`</code>
<code>iot:GetRetainedMessage</code>	<code>topic</code>	Un tema de mensaje conservado.	<code>arn:aws:iot:us-east-1:123456789012:topic/myTopicName</code>
<code>iot:GetThingShadow</code>	<code>thing</code>	El nombre del objeto	<code>arn:aws:iot:us-east-1:123456789012:thing/thingOne</code>
<code>iot>ListRetainedMessages</code>	<code>topic</code>	Todos	<code>*</code>
<code>iot:Publish</code>	<code>topic</code>	Cadena de temas	<code>arn:aws:iot:us-east-1:123456789012:topic/myTopicName</code>
<code>iot:Receive</code>	<code>topic</code>	Cadena de temas	<code>arn:aws:iot:us-east-1:123456789012:topic/myTopicName</code>
<code>iot:RetainPublish</code>	<code>topic</code>	Tema para publicar con el conjunto de indicadores RETAIN.	<code>arn:aws:iot:us-east-1:123456789012:topic/myTopicName</code>

Acción	Tipo de recurso	Nombre del recurso	Ejemplo de ARN
iot:StartNextPendingJobExecution		El nombre del objeto	arn:aws:iot:us-east-1:123456789012:thing/thingOne
iot:Subscribe	topicfilter	Una cadena de filtro de temas	arn:aws:iot:us-east-1:123456789012:topicfilter/myTopicFilter
iot:UpdateJobExecution	thing	El nombre del objeto	arn:aws:iot:us-east-1:123456789012:thing/thingOne
iot:UpdateThingShadow	thing	El nombre de la cosa y el nombre de la sombra, si corresponde	arn:aws:iot:us-east-1:123456789012:thing/thingOne arn:aws:iot:us-east-1:123456789012:thing/thingOne/shadowOne
iot:AssumeRoleWithCertificate		Alias de rol que apunta a un ARN de rol	arn:aws:iot:us-east-1:123456789012:rolealias CredentialProviderRole_alias

Variables de las políticas de AWS IoT Core

AWS IoT Core define las variables de política que se pueden utilizar en las políticas de AWS IoT Core del bloque `Resource` o `Condition`. Cuando se evalúa una política, las variables de política se sustituyen por valores reales. Por ejemplo, si un dispositivo se ha conectado al agente de mensajes de AWS IoT Core con un ID de cliente de 100-234-3456, la variable de política `iot:ClientId` se sustituye en el documento de política por 100-234-3456.

AWS IoT Core las políticas pueden utilizar caracteres comodín y seguir una convención similar a la de las políticas de IAM. Inserción de un*(asterik) de la cadena se puede tratar como un comodín, coincidiendo con cualquier carácter. Por ejemplo, puede utilizar*para describir varios nombres de temas MQTT en el `Resource` atributo de una política. Los personajes+y#se tratan como cadenas literales en una política. Para obtener un ejemplo de política que muestra cómo utilizar comodines, consulte [Políticas para clientes MQTT \(p. 351\)](#).

También puede utilizar variables de política predefinidas con valores fijos para representar caracteres que, de otro modo, tendrían un significado especial. Estos caracteres especiales incluyen\$(*)\$,\$(?)\$, y\$(\\$)\$. Para obtener más información acerca de las variables de políticas y los caracteres especiales, consulte [Elementos de la política de IAM: Variables y etiquetas](#) y [Creación de una condición con varias claves o valores](#).

Temas

- [Variables de política de AWS IoT Core básicas \(p. 337\)](#)
- [Variables de la política de objeto \(p. 339\)](#)
- [Variables de política de AWS IoT Core de certificado X.509 \(p. 339\)](#)

Variables de política de AWS IoT Core básicas

AWS IoT Core define las siguientes variables de política básicas:

- **iot:ClientId**: el ID de cliente que se utiliza para conectarse con el agente de mensajes de AWS IoT Core.
- **aws:SourceIp**: la dirección IP del cliente conectado con el agente de mensajes de AWS IoT Core.

La siguiente política de AWS IoT Core muestra una política que utiliza variables 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}"  
            ]  
        }  
    ]  
}
```

En estos ejemplos, \${iot:ClientId} se sustituirá por el ID del cliente conectado al agente de mensajes de AWS IoT Core cuando se evalúa la política. Cuando utiliza variables de la política como \${iot:ClientId}, puede abrir de forma accidental el acceso a temas que no quería incluir. Por ejemplo, si utiliza una política que utiliza \${iot:ClientId} para especificar un filtro de temas:

```
{  
    "Effect": "Allow",  
    "Action": ["iot:Subscribe"],  
    "Resource": [  
        "arn:aws:iot:us-east-1:123456789012:topicfilter/my/${iot:ClientId}/topic"  
    ]  
}
```

Un cliente puede conectarse usando + como ID de cliente. Esto puede permitir al usuario suscribirse a cualquier tema que coincida con el filtro de temas my/+ topic. Como protección contra estas deficiencias de seguridad, utilice la acción de política iot:Connect para controlar los ID de cliente que pueden conectarse. Por ejemplo, esta política permite conectarse únicamente a los clientes cuyo ID de cliente sea clientid1:

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": ["iot:Connect"],  
            "Resource": [  
                "arn:aws:iot:us-east-1:123456789012:client/clientid1"  
            ]  
        }  
    ]  
}
```

Variables de la política de objeto

Las variables de política de objeto le permiten escribir políticas de AWS IoT Core que concedan o denieguen permisos en función de las propiedades del objeto como el nombre o el tipo de objeto, o los valores de atributo del objeto. Puede usar variables de política de objeto para aplicar la misma política para controlar muchos dispositivos de AWS IoT Core. Para obtener más información acerca del aprovisionamiento de dispositivos, consulte [Aprovisionamiento de dispositivos](#). El nombre de objeto se obtiene a partir del ID de cliente en el mensaje Connect de MQTT que se envía cuando un objeto se conecta con AWS IoT Core.

Tenga en cuenta lo siguiente cuando utilice variables en las políticas de AWS IoT Core.

- Usar [AttachThingPrincipalAPI](#) para asociar certificados o entidades de seguridad (identidades autenticadas de Amazon Cognito) a un objeto.
- Si reemplaza los nombres de objeto con variables de política de objeto, el valor de `clientId` en el mensaje de conexión MQTT o la conexión TLS debe coincidir exactamente con el nombre del objeto.

Las siguientes variables de política de objeto están disponibles:

- `iot:Connection.Thing.ThingName`

Esta variable se resuelve en el nombre del objeto en el registro de AWS IoT Core para el que se evalúa la política. AWS IoT Core utiliza el certificado que presenta el dispositivo cuando se autentica para determinar qué objeto que se va a utilizar para verificar la conexión. Esta variable de política solo está disponible cuando un dispositivo se conecta a través de MQTT o MQTT sobre WebSocket Protocolo.

- `iot:Connection.Thing.ThingTypeName`

Esta variable se resuelve en el tipo de objeto asociado al objeto para el que se evalúa la política. El nombre de objeto se establece en el ID de cliente de la conexión MQTT/WebSocket. Esta variable de política solo está disponible cuando la conexión se establece sobre MQTT o MQTT sobre protocolo WebSocket.

- `iot:Connection.Thing.Attributes[attributeName]`

Esta variable se resuelve en el valor del atributo especificado asociado al objeto para el que se evalúa la política. Un objeto puede tener hasta 50 atributos. Cada atributo estará disponible como variable de política: `iot:Connection.Thing.Attributes[attributeName]` donde `attributeName` es el nombre del atributo. El nombre de objeto se establece en el ID de cliente de la conexión MQTT/ WebSocket. Esta variable de política solo está disponible cuando la conexión se establece sobre MQTT o MQTT sobre WebSocket Protocolo.

- `iot:Connection.Thing.IsAttached`

`iot:Connection.Thing.IsAttached: ["true"]` hace cumplir que solo los dispositivos que están registrados en AWS IoT adjuntado al principal pueden acceder a los permisos dentro de la política. Puede utilizar esta variable para evitar que un dispositivo se conecte a AWS IoT Core si presenta un certificado que no está asociado a un objeto de IoT en la AWS IoT Core Registro. Esta variable tiene valores `true` o `false` que indica que la cosa de conexión está asociada al certificado o a la identidad de Amazon Cognito en el registro mediante [AttachThingPrincipalAPI](#). El nombre de la cosa se toma como ID de cliente.

Variables de política de AWS IoT Core de certificado X.509

Las variables de política de certificado X.509 le permiten escribir políticas de AWS IoT Core que concedan permisos basados en los atributos de certificado X.509. En las secciones siguientes se describe cómo se pueden utilizar estas variables de política de certificado.

CertificateId

En la API [RegisterCertificate](#) el `certificateId` aparece en el cuerpo de la respuesta. Para obtener información sobre su certificado, puede usar el `certificateId` en [DescribeCertificate](#).

Atributos de emisor

Las siguientes variables de política de AWS IoT Core le permiten autorizar o denegar permisos en función de atributos de certificado establecidos por el emisor del 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 sujeto

Las siguientes variables de política de AWS IoT Core le permiten conceder o denegar permisos en función de atributos de sujeto de certificado establecidos por el emisor del 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`

Los certificados X.509 permiten que estos atributos contengan uno o varios valores. De forma predeterminada, las variables de política de cada atributo de varios valores devuelven el primer valor. Por ejemplo, el atributo `Certificate.Subject.Country` podría contener una lista de nombres de países, pero cuando se evalúa en una política, `iot:Certificate.Subject.Country` se reemplaza por el nombre del primer país. Puede solicitar un valor de atributo específico distinto del primero mediante un índice de base uno. Por ejemplo, `iot:Certificate.Subject.Country.1` se sustituye por el nombre del segundo país en el atributo `Certificate.Subject.Country`. Si especifica un valor de índice que

no existe (por ejemplo, si pide un tercer valor cuando el atributo solo tiene dos valores asignados), no se realizará ninguna sustitución y la autorización dará un resultado erróneo. Puede utilizar el sufijo `.List` en el nombre de la variable de política para especificar todos los valores del atributo.

Registered devices (2)

Para los dispositivos registrados como objetos en el registro de AWS IoT Core, la siguiente política permite que los clientes con un nombre de objeto registrado en el registro de AWS IoT Core se conecten, pero limita el derecho a publicar en un tema específico del nombre de objeto a aquellos clientes con certificados cuyo atributo `Certificate.Subject.Organization` se haya establecido en "Example Corp" o en "AnyCompany". Esta restricción se lleva a cabo mediante un campo "Condition" que especifica una condición que ha de cumplirse para permitir la acción anterior. En este caso, la condición es que el atributo `Certificate.Subject.Organization` asociado al certificado incluya uno de los valores de la lista:

```
{  
    "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 los dispositivos no registrados como objetos en el registro de AWS IoT Core, la siguiente política concede permiso para conectarse a AWS IoT Core con los ID de cliente `client1`, `client2` y `client3`, pero limita el derecho a publicar en un tema específico del id de cliente a aquellos clientes con certificados cuyo atributo `Certificate.Subject.Organization` se haya establecido en "Example Corp" o en "AnyCompany". Esta restricción se lleva a cabo mediante un campo "Condition" que especifica una condición que ha de cumplirse para permitir la acción anterior. En este caso, la condición es que el atributo `Certificate.Subject.Organization` asociado al certificado incluya uno de los valores de la lista:

```
{  
    "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 nombre alternativo del emisor

Las variables de política de AWS IoT Core siguientes le permiten conceder o denegar permisos en función de los atributos de nombre alternativo del emisor establecidos por el emisor del 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 nombre alternativo de sujeto

Las variables de política de AWS IoT Core siguientes le permiten conceder o denegar permisos en función de los atributos de nombre alternativo del tema establecidos por el emisor del 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`

Otros atributos

Puede utilizar `iot:Certificate.SerialNumber` para permitir o denegar el acceso a recursos de AWS IoT Core en función del número de serie de un certificado. La variable de política

`iot:Certificate.AvailableKeys` contiene el nombre de todas las variables de política de certificado que contienen valores.

Limitaciones aplicables a las variables de política de certificado X.509

Las siguientes limitaciones se aplican a las variables de política de certificado X.509:

Caracteres comodín

Si los atributos de certificado contienen caracteres comodín, la variable de política no se sustituirá por el valor de atributo de certificado y el texto `#{policy-variable}` figurará en el documento de la política. Esto puede producir un error de autorización. Se pueden utilizar los siguientes caracteres comodín: *, \$, +, ? y #.

Campos de matriz

Los atributos de certificado que contienen matrices se limitan a cinco elementos. No se tendrán en cuenta los elementos adicionales.

Longitud de cadena

Todos los valores de cadena están limitados a 1024 caracteres. Si un atributo de certificado contiene una cadena de más de 1024 caracteres, la variable de política no se sustituirá por el valor de atributo de certificado y el texto `#{policy-variable}` figurará en el documento de la política. Esto puede producir un error de autorización.

Caracteres especiales

Cualquier carácter especial, como , , " , \, +, =, <, > y ; debe tener el prefijo de una barra invertida (\) cuando se utiliza en una variable de política. Por ejemplo, Amazon Web Services O=Amazon.com Inc. L=Seattle ST=Washington C=US se convierte en Amazon Web Service O\=Amazon.com Inc. L\=Seattle ST\=Washington C\=US.

Prevención del suplente confuso entre servicios

El problema de la sustitución confusa es una cuestión de seguridad en la que una entidad que no tiene permiso para realizar una acción puede obligar a una entidad con más privilegios a realizar la acción. En AWS, la suplantación entre servicios puede dar lugar al problema del suplente confuso. La suplantación entre servicios puede producirse cuando un servicio (el servicio que lleva a cabo las llamadas) llama a otro servicio (el servicio al que se llama). El servicio que lleva a cabo las llamadas se puede manipular para utilizar sus permisos para actuar en función de los recursos de otro cliente de una manera en la que no debería tener permiso para acceder. Para evitarlo, AWS proporciona herramientas que lo ayudan a proteger sus datos para todos los servicios con entidades principales de servicio a las que se les ha dado acceso a los recursos de su cuenta.

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

La forma más eficaz de protegerse contra el problema del suplente confuso es utilizar `aws:SourceArn` clave de contexto de condición global con el nombre de recurso de Amazon (ARN) completo. Para AWS IoT, `aws:SourceArn` debe cumplir con el formato de: `arn:aws:iot:region:account-id:*`. Asegúrese de que el `region` coincide con su AWS IoT Región y el `account-id` coincide con el ID de cuenta de cliente.

El siguiente ejemplo muestra cómo evitar el problema del suplente confuso utilizando `aws:SourceArn` y `aws:SourceAccount` Claves de contexto de condición globales de en AWS IoT política de confianza de rol.

```
{  
    "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:/*"  
                }  
            }  
        }  
    ]  
}
```

Ejemplos de políticas de AWS IoT Core

Las políticas de ejemplo de esta sección ilustran los documentos de política utilizados para realizar tareas comunes en AWS IoT Core. Puede utilizarlos como ejemplos para empezar a partir de la creación de las políticas para sus soluciones.

En los ejemplos de esta sección se utilizan estos elementos de política:

- the section called “Acciones de política de AWS IoT Core” (p. 334)
- the section called “Recursos de acción de AWS IoT Core” (p. 336)
- the section called “Ejemplos de políticas basadas en identidad” (p. 408)
- the section called “Variables de política de AWS IoT Core básicas” (p. 337)
- the section called “Variables de política de AWS IoT Core de certificado X.509” (p. 339)

Ejemplos de políticas de esta sección:

- [Ejemplos de política de conexión \(p. 344\)](#)
- [Ejemplos de política de publicación/suscripción \(p. 351\)](#)
- [Ejemplos de políticas de conexión y publicación \(p. 365\)](#)
- [Ejemplos de políticas de mensajes conservados \(p. 366\)](#)
- [Ejemplos de políticas de certificado \(p. 368\)](#)
- [Ejemplos de políticas de objeto \(p. 372\)](#)
- [Ejemplo de política de trabajo básica \(p. 372\)](#)

Ejemplos de política de conexión

La siguiente política concede permiso para conectarse a AWS IoT Core con el ID de cliente `client1`:

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

```
        ],
        "Resource": [
            "arn:aws:iot:us-east-1:123456789012:client/client1"
        ]
    }
}
```

La siguiente política deniega el permiso a ID de clientes `client1` y `client2` para conectar a AWS IoT Core, a la vez que permite a los dispositivos conectarse mediante un ID de cliente que coincida con el nombre de un objeto registrado en el registro de 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}"
            ]
        }
    ]
}
```

Ejemplos de políticas de sesiones persistentes de MQTT

`connectAttributes` le permiten especificar qué atributos desea utilizar en el mensaje de conexión en sus políticas de IAM, tales como `PersistentConnect` y `LastWill`. Para obtener más información, consulte [Uso de ConnectAttributes \(p. 94\)](#)

La siguiente política permite conectarse con `PersistentConnect` característica:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:Connect"
            ],
            "Resource": "*",
            "Condition": {
                "ForAllValues:StringEquals": {
                    "iot:ConnectAttributes": [
                        "PersistentConnect"
                    ]
                }
            }
        }
    ]
}
```

```
    ]  
}
```

La siguiente política no permite `PersistentConnect`, se permiten otras funciones:

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "iot:Connect"  
            ],  
            "Resource": "*",  
            "Condition": {  
                "ForAllValues:StringNotEquals": {  
                    "iot:ConnectAttributes": [  
                        "PersistentConnect"  
                    ]  
                }  
            }  
        }  
    ]  
}
```

La política anterior también se puede expresar utilizando `StringEquals`, se permite cualquier otra función, incluida la nueva función:

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "iot:Connect"  
            ],  
            "Resource": "*"  
        },  
        {  
            "Effect": "Deny",  
            "Action": [  
                "iot:Connect"  
            ],  
            "Resource": "*",  
            "Condition": {  
                "ForAnyValue:StringEquals": {  
                    "iot:ConnectAttributes": [  
                        "PersistentConnect"  
                    ]  
                }  
            }  
        }  
    ]  
}
```

La siguiente política permite conectarse con ambos `PersistentConnect`y `LastWill`, no se permite ninguna otra función nueva:

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

```
"Action": [
    "iot:Connect"
],
"Resource": "*",
"Condition": {
    "ForAllValues:StringEquals": {
        "iot:ConnectAttributes": [
            "PersistentConnect",
            "LastWill"
        ]
    }
}
]
```

La siguiente política permite la conexión limpia de los clientes con o sin `LastWill`, no se permitirán otras funciones:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:Connect"
            ],
            "Resource": "*",
            "Condition": {
                "ForAllValues:StringEquals": {
                    "iot:ConnectAttributes": [
                        "LastWill"
                    ]
                }
            }
        }
    ]
}
```

La siguiente directiva solo permite conectarse mediante funciones predeterminadas:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:Connect"
            ],
            "Resource": "*",
            "Condition": {
                "ForAllValues:StringEquals": {
                    "iot:ConnectAttributes": [
                    ]
                }
            }
        }
    ]
}
```

La siguiente política permite conectarse solo con `PersistentConnect`, se permite cualquier nueva función siempre que la conexión utilice `PersistentConnect`:

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "iot:Connect"  
            ],  
            "Resource": "*",  
            "Condition": {  
                "ForAnyValue:StringEquals": {  
                    "iot:ConnectAttributes": [  
                        "PersistentConnect"  
                    ]  
                }  
            }  
        }  
    ]  
}
```

La siguiente política indica que la conexión debe tener ambos `PersistentConnect`y `LastWill`, no se permite ninguna nueva función:

```
{  
    "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": [  
                        "PersistentConnect"  
                    ]  
                }  
            }  
        }  
    ]  
}
```

```
        "LastWill"
    ]
}
},
{
    "Effect": "Deny",
    "Action": [
        "iot:Connect"
    ],
    "Resource": "*",
    "Condition": {
        "ForAllValues:StringEquals": {
            "iot:ConnectAttributes": [
                "LastWill"
            ]
        }
    }
]
}
```

La siguiente política no debe tener `PersistentConnect` pero puede tener `LastWill`, no se permite ninguna otra función nueva:

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

La siguiente política permite conectarse solo por clientes que tienen un `LastWill` con tema "`my/lastwill/topicName`", se permite cualquier característica siempre que utilice la `LastWill` tema:

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

La siguiente política solo permite una conexión limpia mediante una conexión específica `LastWillTopic`, se permite cualquier característica siempre que utilice la `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)

La siguiente política concede permiso a un dispositivo para conectarse utilizando su nombre de objeto como ID de cliente y para suscribirse al filtro de temas `my/topic/filter`. El dispositivo debe estar registrado en AWS IoT Core. Cuando el dispositivo se conecta a AWS IoT Core, debe proporcionar el certificado asociado con el objeto de IoT en el registro de 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 los dispositivos no registrados como objetos en el registro de AWS IoT Core, la siguiente política concede permiso para conectarse utilizando el ID de cliente `client1` y para suscribirse al filtro de temas `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"  
            ]  
        }  
    ]  
}
```

Ejemplos de política de publicación/suscripción

La política que utilice dependerá de cómo se conecte con AWS IoT Core. Puedes conectarte a AWS IoT Core mediante un cliente MQTT, HTTP o WebSocket. Al conectarse con un cliente MQTT, se autenticará con un certificado X.509. Cuando te conectas a través de HTTP o el protocolo WebSocket, se autenticará con Signature Version 4 y Amazon Cognito.

Políticas para clientes MQTT

Para describir varios nombres de temas y filtros de temas en el `resource` atributo de una política, utilice el `*y?` carácter comodín en lugar de los caracteres comodín MQTT.

MQTT y AWS IoT Core las políticas tienen caracteres comodín diferentes y deben elegirse tras una cuidadosa consideración. En MQTT, los caracteres comodín + y # se utilizan en [Filtros de temas MQTT](#) para suscribirse a varios nombres de temas. AWS IoT Core usa de políticas de * y ? como caracteres comodín y seguir las convenciones de [Políticas de IAM](#). En un documento de políticas, la * representa cualquier combinación de caracteres y un signo de interrogación ? representa un único carácter. En los documentos de políticas, los caracteres comodín MQTT, + y # se tratan como esos personajes sin ningún significado especial.

Al elegir caracteres comodín para utilizarlos en un documento de política, tenga en cuenta que la * carácter no se limita a un solo nivel de tema, ya que el + personaje se encuentra en un filtro de temas MQTT. Para ayudar a restringir una especificación comodín a un único nivel de filtro de temas MQTT, considere utilizar varios ? caracteres. Para obtener más información sobre el uso de caracteres comodín en un recurso de política y más ejemplos de lo que coinciden, consulte [Uso de comodines en ARN de recursos](#).

En la siguiente tabla se muestran los distintos caracteres comodín utilizados en MQTT y AWS IoT Core políticas para clientes MQTT.

Carácter comodín	Es un carácter comodín MQTT	Ejemplo en MQTT	Es AWS IoT Core carácter comodín de política	Ejemplo de en AWS IoT Core políticas para clientes MQTT
#	Sí	some/#	No	N/A
+	Sí	some/+/topic	No	N/A
*	No	N/A	Sí	topicfilter/some/*/topic
?	No	N/A	Sí	topic/some/?????/topic topicfilter/some/sensor???/topic

La política siguiente permite que un dispositivo publique en todos los subtemas que comiencen con el mismo nombre de objeto.

Registered devices (5)

Para los dispositivos registrados como objetos en el registro de AWS IoT Core, la siguiente política concede permiso para conectarse a AWS IoT Core utilizando un ID de cliente que coincide con el nombre de objeto y para publicar en cualquier tema que tenga como prefijo el nombre del objeto:

```
{
    "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"
            ],
            "TopicFilter": [
                "topicfilter/some/*/topic"
            ]
        }
    ]
}
```

```
        "iot:Publish"
    ],
    "Resource": [
        "arn:aws:iot:us-east-1:123456789012:topic/
${iot:Connection.Thing.ThingName}/*"
    ]
}
}
```

Unregistered devices (5)

Para los dispositivos no registrados como objetos en el registro de AWS IoT Core, la siguiente política concede permiso para conectarse a AWS IoT Core utilizando el ID de cliente `client1`, `client2` o `client3` y para publicar en cualquier tema que tenga como prefijo el ID de 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}/*"
            ]
        }
    ]
}
```

También puede utilizar el carácter comodín `*` al final de un filtro de tema. El uso de caracteres comodín podría dar lugar a la concesión de privilegios no deseados, por lo que solo deben utilizarse teniendo especial cuidado. Una situación en la que podrían ser útiles es cuando los dispositivos deben suscribirse a mensajes con muchos temas distintos (por ejemplo si un dispositivo debe suscribirse a informes de sensores de temperatura en varias ubicaciones).

Registered devices (6)

Para los dispositivos registrados como objetos en el registro de AWS IoT Core, la siguiente política concede permiso para conectarse a AWS IoT Core utilizando como ID de cliente el nombre de objeto del dispositivo y para suscribirse a un tema que tenga como prefijo el nombre del objeto, seguido de `room`, seguido de cualquier cadena. (Se espera que estos temas sean, por ejemplo `thing1/room1`, `thing1/room2`, etc.):

```
{
    "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 los dispositivos no registrados como objetos en el registro de AWS IoT Core, la siguiente política concede permiso para conectarse a AWS IoT Core utilizando los ID de cliente `client1`, `client2` y `client3`, y para suscribirse a un tema que tenga como prefijo el ID de cliente, seguido de `room`, seguido de cualquier cadena. (Se espera que estos temas sean, por ejemplo `client1/room1`, `client1/room2`, etc.):

```
{
    "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/${iot:ClientId}/room*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iot:Publish"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:topicfilter/${iot:ClientId}/room*"
            ]
        }
    ]
}
```

```
        "Action": [
            "iot:Receive"
        ],
        "Resource": [
            "arn:aws:iot:us-east-1:123456789012:topic/${iot:ClientId}/room*"
        ]
    }
}
```

Al especificar filtros de temas enAWS IoT Corepolíticas para clientes MQTT, caracteres comodín MQTT+/#se tratan como cadenas literales. Su uso puede dar lugar a un comportamiento inesperado.

Registered devices (4)

Para dispositivos registrados como elementos en elAWS IoT Coreregistro, la siguiente política concede permiso para conectarse aAWS IoT Corecon el ID de cliente que coincide con el nombre del objeto y para suscribirse al filtro de temassome/+topicSolo la . Nota entopicfilter/some/+/topicdel ARN del recurso, + se trata como una cadena literal enAWS IoT Corepolíticas para clientes MQTT, lo que significa que solo la cadenasome/+topiccoincide con el filtro de temas.

```
{
    "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 elementos en elAWS IoT Coreregistro, la siguiente política concede permiso para conectarse aAWS IoT Corecon el ID de cliente que coincide con el nombre del objeto y para suscribirse al filtro de temassome/*topic. Nota entopicfilter/some/*topicdel ARN del recurso, * se trata como un carácter comodín enAWS IoT Corepolíticas para clientes MQTT, lo que significa que cualquier cadena del nivel que contiene el carácter coincide con el filtro de temas. (Se espera que estos temas sean, por ejemplo,some/string1/topic,some/string2/topic, y así sucesivamente)

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:Connect"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:topicfilter/some/*topic"
            ]
        }
    ]
}
```

```

        ],
        "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 no registrados como elementos en elAWS IoT Coreregistro, la siguiente política concede permiso para conectarse aAWS IoT Corecon ID de clienteclient1y suscribirse al filtro de temassome/+/*topicSolo la . Nota entopicfilter/some/+/*topicdel ARN del recurso, + se trata como una cadena literal enAWS IoT Corepolíticas para clientes MQTT, lo que significa que solo la cadenasome/+/*topiccoincide con el filtro de temas.

```
{
    "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 no registrados como elementos en elAWS IoT Coreregistro, la siguiente política concede permiso para conectarse aAWS IoT Corecon ID de clienteclient1y suscribirse al filtro de temassome/+/*topic. Nota entopicfilter/some/*/*topicdel ARN del recurso, * se trata como un carácter comodín enAWS IoT Corepolíticas para clientes MQTT, lo que significa que cualquier cadena del nivel que contiene el carácter coincide con el filtro de temas. (Se espera que estos temas sean, por ejemplo,some/string1/topic,some/string2/topic, y así sucesivamente)

```
{
    "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

Comodines de MQTT+<#se tratan como cadenas literales en unAWS IoT Corepolítica para clientes MQTT. Para especificar caracteres comodín en nombres de temas y filtros de temas enAWS IoT Corepolíticas para clientes de MQTT, debe utilizar*.

Registered devices (7)

Para los dispositivos registrados como objetos en el registro de AWS IoT Core, la siguiente política concede permiso para conectarse a AWS IoT Core utilizando el nombre de objeto del dispositivo como ID de cliente y para suscribirse a los temas `my/topic` y `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 los dispositivos no registrados como objetos en el registro de AWS IoT Core, la siguiente política concede permiso para conectarse a AWS IoT Core utilizando el ID de cliente `client1` y para suscribirse a los temas `my/topic` y `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 los dispositivos registrados como objetos en el registro de AWS IoT Core, la siguiente política concede permiso para conectarse a AWS IoT Core utilizando el nombre de objeto del dispositivo como ID de cliente y para suscribirse a un tema único para dicho nombre de objeto o ID de 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 los dispositivos no registrados como objetos en el registro de AWS IoT Core, la siguiente política concede permiso para conectarse a AWS IoT Core utilizando el ID de cliente `client1` y para publicar en un tema único para dicho 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 los dispositivos registrados como objetos en el registro de AWS IoT Core, la siguiente política concede permiso para conectarse a AWS IoT Core utilizando como ID de cliente el nombre de objeto del dispositivo y para publicar en cualquier tema que lleve como prefijo el nombre del objeto o cliente excepto en el caso de un tema que termine por 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 los dispositivos no registrados como objetos en el registro de AWS IoT Core, la siguiente política concede permiso para conectarse a AWS IoT Core utilizando como ID de cliente `client1` y `client1`, y para publicar en cualquier tema que lleve como prefijo el ID de cliente utilizado para conectarse, excepto en el caso de un tema que termine por `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 los dispositivos registrados como objetos en el registro de AWS IoT Core, la siguiente política concede permiso para conectarse a AWS IoT Core utilizando el nombre de objeto del dispositivo como ID de cliente. El dispositivo puede suscribirse al tema `my/topic`, pero no se puede publicar en el `thing-name /bar` donde `Nombre de cosa` es el nombre de la cosa de IoT a la que se conecta AWS IoT Core:

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "iot:Connect"  
            ],  
            "Resource": [  
                "arn:aws:iot:us-east-1:123456789012:device/thing-name"  
            ]  
        }  
    ]  
}
```

```
        "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 los dispositivos no registrados como objetos en el registro de AWS IoT Core, la siguiente política concede permiso para conectarse a AWS IoT Core utilizando el ID de cliente `client1` y para suscribirse al tema `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"  
            ]  
        }  
    ]  
}
```

Las variables de política de objeto también se sustituyen cuando se asocia un certificado o Amazon Cognito Identity autenticado a un objeto. La siguiente política concede permiso para conectarse con AWS IoT Core con el ID del cliente `client1` y para publicar y recibir el tema `iotmonitor/provisioning/987654321098`. También permite que el titular del certificado se suscriba este tema.

```
{  
    "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 de HTTP y WebSocket

Las identidades de Amazon Cognito se pueden autenticar o no autenticar. Las identidades autenticadas pertenecen a los usuarios que se han autenticado mediante un proveedor de identidad compatible. En cuanto a las identidades sin autenticar normalmente corresponden a usuarios invitados que no se autentiquen con un proveedor de identidades. Amazon Cognito proporciona un identificador único yAWScredentials para admitir identidades sin autenticar.

Para las siguientes operaciones,AWS IoT CoreutilizaAWS IoT Corepolíticas asociadas a las identidades de Amazon Cognito (a través delAttachPolicyAPI) para reducir el ámbito de los permisos asociados al grupo de Amazon Cognito Identity con identidades autenticadas.

- iot:Connect
- iot:Publish
- iot:Subscribe
- iot:Receive
- iot:GetThingShadow
- iot:UpdateThingShadow
- iot:DeleteThingShadow

Esto significa que una Amazon Cognito Identity necesita el permiso de la política de rol de IAM asociada al grupo y alAWS IoT Corepolítica adjunta a Amazon Cognito Identity a través delAWS IoT Core AttachPolicyAPI.

Los usuarios autenticados y no autenticados son tipos de identidad diferentes. Si no adjunta una AWS IoT política de Amazon Cognito Identity, un usuario autenticado falla la autorización en AWS IoT y no tiene acceso a AWS IoT recursos y acciones.

Note

Para otras AWS IoT Core operaciones o identidades sin autenticar, AWS IoT Core reduce el ámbito de los permisos asociados al rol del grupo de identidades de Amazon Cognito. Para las identidades autenticadas y sin autenticar, esta es la política más permisiva que recomendamos asociar al rol del grupo de Amazon Cognito.

HTTP

Para permitir que identidades de Amazon Cognito sin autenticar publiquen mensajes sobre HTTP en un tema específico de Amazon Cognito Identity, asocie la política de IAM siguiente al rol de grupo de Amazon Cognito Identity:

```
{  
    "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 usuarios autenticados, adjunte la política anterior al rol de grupo de Amazon Cognito Identity y a Amazon Cognito Identity mediante la AWS IoT Core [AttachPolicy API](#).

Note

Al autorizar identidades de Amazon Cognito, AWS IoT Core tiene en cuenta ambas políticas y concederá los mínimos privilegios especificados. Solo se permite una acción si ambas políticas permiten la acción solicitada. Si una de ellas no permite una acción, esa acción no se autoriza.

MQTT

Para permitir que las identidades de Amazon Cognito no autenticadas publiquen mensajes MQTT en WebSocket en un tema específico de Amazon Cognito Identity en su cuenta, asocie la política de IAM siguiente al rol del grupo de Amazon Cognito Identity:

```
{  
    "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/*"]  
        }  
    ]  
}
```

```
        "iot:Connect"
    ],
    "Resource": ["arn:aws:iot:us-east-1:123456789012:client/${cognito-identity.amazonaws.com:sub}"]
}
]
```

Para permitir usuarios autenticados, adjunte la política anterior al rol de grupo de Amazon Cognito Identity y a Amazon Cognito Identity mediante laAWS IoT Core [AttachPolicyAPI](#).

Note

Al autorizar identidades de Amazon Cognito,AWS IoT Coretiene en cuenta ambas y concederá los mínimos privilegios especificados. Solo se permite una acción si ambas políticas permiten la acción solicitada. Si una de ellas no permite una acción, esa acción no se autoriza.

Ejemplos de políticas de recepción

Registered devices (11)

Para los dispositivos registrados en el registro de AWS IoT Core, la siguiente política concede permiso para conectarse a AWS IoT Core con un ID de cliente que coincide con el nombre de objeto y para suscribirse y recibir mensajes en el tema my/topic:

```
{
    "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 los dispositivos no registrados en el registro de AWS IoT Core, la siguiente política concede permiso para conectarse a AWS IoT Core con un ID de cliente `client1` y para suscribirse y recibir mensajes en un tema:

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

Ejemplos de políticas de conexión y publicación

Para los dispositivos registrados como objetos en el registro de AWS IoT Core, la siguiente política concede permiso para conectarse a AWS IoT Core con un ID de cliente que coincide con el nombre de objeto y limita las publicaciones del dispositivo a un tema de MQTT específico del ID de cliente o nombre de objeto. Para que una conexión se realice correctamente, el nombre del objeto debe estar registrado en el registro de AWS IoT Core y se debe autenticar utilizando una identidad o entidad principal asociada al objeto:

```
{  
    "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 los dispositivos no registrados como objetos en el registro de AWS IoT Core, la siguiente política concede permiso para conectarse a AWS IoT Core con un ID de cliente `client1` y limita las publicaciones del dispositivo a un tema de MQTT específico del ID de cliente:

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

Ejemplos de políticas de mensajes conservados

Uso de [mensajes retenidos](#) (p. 88) requiere políticas específicas. Los mensajes retenidos son mensajes MQTT publicados con la marca RETAIN establecida y almacenada por AWS IoT Core. En esta sección se presentan ejemplos de políticas que permiten el uso común de los mensajes retenidos.

En esta sección:

- [Política para conectar y publicar mensajes retenidos \(p. 366\)](#)
- [Política de conexión y publicación de mensajes Will retenidos \(p. 367\)](#)
- [Política para listar y obtener mensajes retenidos \(p. 367\)](#)

Política para conectar y publicar mensajes retenidos

Para que un dispositivo publique mensajes retenidos, el dispositivo debe poder conectarse, publicar (cualquier mensaje MQTT) y publicar mensajes retenidos de MQTT. La siguiente política concede estos permisos para el tema `device/sample/configuration` del cliente `device1`. Para ver otro ejemplo que concede permiso para conectarse, consulte [the section called “Ejemplos de políticas de conexión y publicación” \(p. 365\)](#).

```
{  
    "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 de conexión y publicación de mensajes Will retenidos

Los clientes pueden configurar un mensaje que AWS IoT Core publicará cuando el cliente se desconecte inesperadamente. MQTT llama a ese mensaje como [Voluntadmensaje](#). Un cliente debe tener agregada una condición adicional a su permiso de conexión para incluirlo.

El siguiente documento de política concede permiso a todos los clientes para conectarse y publicar un mensaje Will, identificado por su tema, `will`, que AWS IoT Core también se conservará.

```
{  
    "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 y obtener mensajes retenidos

Los servicios y las aplicaciones pueden acceder a los mensajes retenidos sin necesidad de admitir un cliente MQTT llamando [ListRetainedMessages](#) y [GetRetainedMessage](#). Los servicios y aplicaciones que llaman a estas acciones deben autorizarse mediante una política como el siguiente ejemplo.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "iot>ListRetainedMessages"  
            ],  
            "Resource": [  
                "*"  
            ]  
        },  
        {  
            "Effect": "Allow",  
            "Action": [  
                "iot:GetRetainedMessage"  
            ],  
            "Resource": [  
                "*"  
            ]  
        }  
    ]  
}
```

```
        ],
        "Resource": [
            "arn:aws:iot:us-east-1:123456789012:topic/foo"
        ]
    }
}
```

Ejemplos de políticas de certificado

Para los dispositivos registrados en el registro de AWS IoT Core, la siguiente política concede permiso para conectarse a AWS IoT Core con un ID de cliente que coincide con un nombre de objeto y para publicar en un tema cuyo nombre sea igual al `certificateId` del certificado que el dispositivo utilizó para autenticarse:

```
{
    "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 los dispositivos no registrados en el registro de AWS IoT Core, la siguiente política concede permisos para conectarse a AWS IoT Core con los ID de cliente `client1`, `client2` y `client3`, y para publicar en un tema cuyo nombre sea igual al `certificateId` del certificado que el dispositivo utilizó para autenticarse:

```
{
    "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 los dispositivos registrados en el registro de AWS IoT Core, la siguiente política concede permiso para conectarse a AWS IoT Core con un ID de cliente que coincide con el nombre de objeto y para publicar en un tema cuyo nombre sea igual al campo CommonName del sujeto del certificado que el dispositivo utilizó para autenticarse:

```
{  
    "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

En este ejemplo, el nombre común del sujeto del certificado se utiliza como identificador de tema, partiendo del supuesto de que el nombre común del sujeto es único para cada certificado registrado. Si los certificados se comparten entre varios dispositivos, el nombre común del asunto es el mismo para todos los dispositivos que comparten este certificado, lo que permite publicar privilegios en el mismo tema desde varios dispositivos (no se recomienda).

Para los dispositivos no registrados en el registro de AWS IoT Core, la siguiente política concede permiso para conectarse a AWS IoT Core con los ID de cliente `client1`, `client2` y `client3`, y para publicar en un tema cuyo nombre sea igual al campo CommonName del sujeto del certificado que el dispositivo utilizó para autenticarse:

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

```
        "arn:aws:iot:us-east-1:123456789012:client/client2",
        "arn:aws:iot:us-east-1:123456789012:client/client3"
    ]
}
}
```

Note

En este ejemplo, el nombre común del sujeto del certificado se utiliza como identificador de tema, partiendo del supuesto de que el nombre común del sujeto es único para cada certificado registrado. Si los certificados se comparten entre varios dispositivos, el nombre común del asunto es el mismo para todos los dispositivos que comparten este certificado, lo que permite publicar privilegios en el mismo tema desde varios dispositivos (no se recomienda).

Para los dispositivos registrados en el registro de AWS IoT Core, la siguiente política concede permiso para conectarse a AWS IoT Core con un ID de cliente que coincide con un nombre de objeto y para publicar en un tema cuyo nombre tenga como prefijo `admin/` cuando el certificado utilizado para autenticar el dispositivo tenga el campo `Subject.CommonName.2` definido en `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 los dispositivos no registrados en el registro de AWS IoT Core, la siguiente política concede permiso para conectarse a AWS IoT Core con los ID de cliente `client1`, `client2` y `client3`, y para publicar en un tema cuyo nombre tenga como prefijo `admin/` cuando el certificado utilizado para autenticar el dispositivo tenga el campo `Subject.CommonName.2` definido en `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"
            ]
        }
    ]
}
```

```

        "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 los dispositivos registrados en el registro de AWS IoT Core, la siguiente política permite a un dispositivo utilizar su nombre de objeto para publicar en un tema específico que consiste en `admin/` seguido del `ThingName` cuando el certificado utilizado para autenticar el dispositivo tenga alguno de sus campos `Subject.CommonName` establecido en `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 los dispositivos no registrados en el registro de AWS IoT Core, la siguiente política concede permiso para conectarse a AWS IoT Core con los ID de cliente `client1`, `client2` y `client3`, y para publicar en el tema `admin` cuando el certificado utilizado para autenticar el dispositivo tenga alguno de sus campos `Subject.CommonName` definido en `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"
        }
    }
}
]
```

Ejemplos de políticas de objeto

La siguiente política permite a un dispositivo conectarse si el certificado utilizado para realizar la autenticación en AWS IoT Core está asociado al objeto cuya política se evalúa:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": ["iot:Connect"],
            "Resource": ["*"],
            "Condition": {
                "Bool": {
                    "iot:Connection.Thing.IsAttached": ["true"]
                }
            }
        }
    ]
}
```

Ejemplo de política de trabajo básica

En este ejemplo se muestran los estados de política requeridos para un objetivo de trabajo que es un único dispositivo para recibir una solicitud de trabajo y comunicar el estado de ejecución del trabajo conAWS IoT.

Reemplazar `us-west-2:57` **EJEMPLO 833** con tu Región de AWS, un carácter de dos puntos (:) y tus 12 dígitos Cuenta de AWS número y, a continuación, reemplace **Nombre de cosa único** con el nombre del recurso de cosa que representa el dispositivo en AWS 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"
    ]
}
]
```

Autorización con identidades de Amazon Cognito

Hay dos tipos de identidades de Amazon Cognito: autenticadas y sin autenticar. Si la aplicación admite identidades de Amazon Cognito sin autenticar, no se realiza ninguna autenticación, por lo que no se sabe quién es el usuario. Para estos usuarios, concede permiso asociando un rol de IAM a un grupo de identidades no autenticado. Le recomendamos que conceda solo acceso a los recursos que desea que estén disponibles para usuarios desconocidos.

Cuando la aplicación admite identidades de Amazon Cognito autenticadas, para autenticar a los usuarios, necesita especificar una política en dos lugares. Adjuntar una política de IAM al grupo de identidad de Amazon Cognito autenticado y adjuntar una AWS IoT Corepolítica de para Amazon Cognito Identity.

Los usuarios autenticados y no autenticados son tipos de identidad diferentes. Si no adjunta una AWS IoT política de Amazon Cognito Identity, un usuario autenticado falla la autorización en AWS IoT y no tiene acceso a AWS IoT recursos y acciones. Para obtener más información acerca de la creación de políticas para identidades de Amazon Cognito, consulte [Ejemplos de política de publicación/suscripción \(p. 351\)](#).

El siguiente ejemplo de aplicaciones web en GitHub muestra cómo incorporar datos adjuntos de directivas a los usuarios autenticados en el proceso de registro y autenticación de usuarios.

- MQTT publicar/suscribirse a la aplicación web React mediante AWS Amplify y la AWS IoT Device SDK para JavaScript
- MQTT publicar/suscribirse a la aplicación web React mediante AWS Amplify, el AWS IoT Device SDK para JavaScript y una función Lambda

Amplify es un conjunto de herramientas y servicios que le ayudan a crear aplicaciones web y móviles que se integran con AWS Servicios de . Para obtener más información acerca de Amplify, consulte [Documentación de Amplify Framework](#).

En ambos ejemplos se realizan los siguientes pasos.

1. Cuando un usuario se registra para obtener una cuenta, la aplicación crea un grupo de usuarios e identidad de Amazon Cognito.
2. Cuando un usuario se autentica, la aplicación crea y adjunta una política a la identidad. Esto otorga al usuario permisos de publicación y suscripción.
3. El usuario puede utilizar la aplicación para publicar y suscribirse a temas de MQTT.

En el primer ejemplo se utiliza la `attachPolicy` API directamente dentro de la operación de autenticación. En el siguiente ejemplo se muestra cómo implementar esta API dentro de una aplicación web de React que utiliza Amplify y el AWS IoT Device SDK para 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);
        }
    });
}
```

Este código aparece en el [AuthDisplay.js](#) que es file.

En el segundo ejemplo se implementa la `attachPolicy` API en una función Lambda. El siguiente ejemplo muestra cómo Lambda utiliza esta 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});
      }
});
```

Este código aparece dentro del `iot.GetPolicy` función en la `app.js` file.

Note

Cuando llamas a la función conAWScredenciales que obtiene a través de los grupos de identidad de Amazon Cognito, el objeto de contexto de la función Lambda contiene un valor para `context.cognito_identity_id`. Para obtener más información, consulte los siguientes temas.

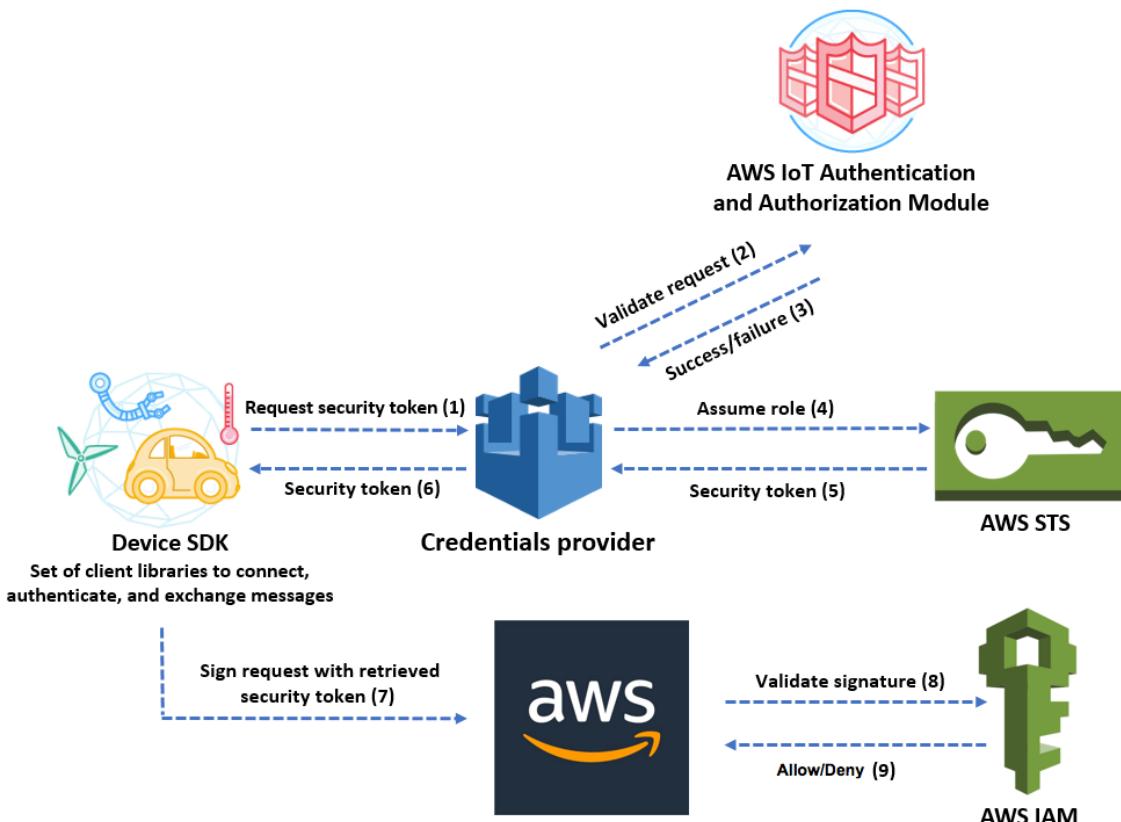
- [AWS LambdaObjeto context de en Node.js](#)
- [AWS LambdaObjeto context de en Python](#)
- [AWS LambdaObjeto context de en Ruby](#)
- [AWS LambdaObjeto context de en Java](#)
- [AWS LambdaObjeto context en Go](#)
- [AWS LambdaObjeto context de en C#](#)
- [AWS LambdaObjeto context de en PowerShell](#)

Autorización de llamadas directas aAWSservicios de utilizandoAWS IoT Coreproveedor de credenciales

Los dispositivos pueden utilizar certificados X.509 para conectarse a AWS IoT Core mediante protocolos de autenticación mutua de TLS. OtroAWSLos servicios no admiten la autenticación de certificados, sino que se pueden llamar a través deAWScredenciales deAWSFormato Signature Version 4. Laalgoritmo Signature Version 4normalmente requiere que el interlocutor tenga un ID de clave de acceso y una clave de acceso secreta.AWS IoT Coretiene un proveedor de credenciales que le permite utilizar la [certificado X.509](#) como identidad de dispositivo única para autenticarseAWSsolicitudes. Así se elimina la necesidad de almacenar un ID de clave de acceso y una clave de acceso secreta en el dispositivo.

El proveedor de credenciales autentica a un intermediario mediante un certificado X.509 y emite un token de seguridad temporal con privilegios limitados. El token se puede utilizar para firmar y autenticar cualquierAWSrequest. Esta forma de autenticar tuAWSlas solicitudes requieren que cree y configure un [AWS Identity and Access Management\(IAM\)](#) roly asocie las políticas de IAM adecuadas para el rol, de modo que el proveedor de credenciales pueda asumir el rol en su nombre. Para obtener más información acerca deAWS IoT Coree IAM, véase [Identity and Access Management en AWS IoT](#) (p. 384).

En el siguiente diagrama se ilustra el flujo de trabajo del proveedor de credenciales.



1. El dispositivo de AWS IoT Core realiza una solicitud HTTPS al proveedor de credenciales para un token de seguridad. La solicitud incluye el certificado X.509 del dispositivo para autenticación.
2. El proveedor de credenciales reenvía la solicitud al módulo de autenticación y autorización de AWS IoT Core para validar el certificado y verificar que el dispositivo tiene permiso para solicitar el token de seguridad.
3. Si el certificado es válido y tiene permiso para solicitar un token de seguridad, el módulo de autenticación y autorización de AWS IoT Core indica que se ha realizado correctamente. De lo contrario, envía una excepción al dispositivo.
4. Despues de validar correctamente el certificado, el proveedor de credenciales invoca a [AWS Security Token Service \(AWS STS\)](#) para asumir el rol de IAM que creó para el mismo.
5. AWS STS devuelve un token de seguridad temporal con privilegios limitados al proveedor de credenciales.
6. El proveedor de credenciales devuelve el token de seguridad al dispositivo.
7. El dispositivo utiliza el token de seguridad para firmar una solicitud con AWSSignature Version 4.
8. El servicio solicitado invoca a IAM para validar la firma y autorizar la solicitud frente a políticas de acceso adjuntas al rol de IAM que creó para el proveedor de credenciales.
9. Si IAM valida la firma correctamente y autoriza la solicitud, la solicitud se realiza correctamente. De lo contrario, IAM envía una excepción.

En la siguiente sección se describe cómo utilizar un certificado para obtener un token de seguridad. Se ha escrito suponiendo que ya ha registrado un dispositivo y [creado y activado su propio certificado](#) para el mismo.

Cómo utilizar un certificado para obtener un token de seguridad

1. Configure el rol de IAM que el proveedor de credenciales asume en nombre de su dispositivo. Adjunte la siguiente política de confianza al rol.

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

Para cadaAWSservicio al que desea llamar, adjunte una política de acceso al rol. El proveedor de credenciales admite las siguientes variables de políticas:

- credentials-iot:ThingName
- credentials-iot:ThingTypeName
- credentials-iot:AwsCertificateId

Cuando el dispositivo proporciona el nombre de objeto en su solicitud a unAWSservicio, añade el proveedor de credenciales credentials-iot:ThingName y credentials-iot:ThingTypeName como variables de contexto del token de seguridad. El proveedor de credenciales proporciona credentials-iot:AwsCertificateId como una variable de contexto incluso si el dispositivo no proporciona el nombre del objeto en la solicitud. Transfiera el nombre del objeto como valor del encabezado de solicitud HTTP x-amzn-iot-thingname.

Estas tres variables solo funcionan para las políticas de IAM, no para las políticas de AWS IoT Core.

2. Asegúrese de que el usuario que realiza el siguiente paso (la creación de un alias de rol) tiene permiso para transferir a AWS IoT Core el rol que se acaba de crear. La siguiente política da ambos iam:GetRole y iam:PassRole permisos para unAWSusuario. El permiso iam:GetRole permite al usuario obtener información acerca del rol que acaba de crear. La iam:PassRole permite al usuario transmitir el rol a otroAWSservice.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {"Effect": "Allow",  
         "Action": [  
             "iam:GetRole",  
             "iam:PassRole"  
         ],  
         "Resource": "arn:aws:iam::your Cuenta de AWS id:role/your role name"  
     }  
]
```

3. Crear un alias de rol de AWS IoT Core. El dispositivo que va a realizar llamadas directas aAWSservicios deben saber qué ARN de rol usar cuando se conecta aAWS IoT Core. La codificación de forma rígida de un ARN de rol no es una buena solución, ya que requiere actualizar el dispositivo cada vez que el ARN del rol cambia. Una solución mejor consiste en utilizar la API CreateRoleAlias para crear un alias de rol que apunte al ARN del rol. Si el ARN del rol cambia, solo tiene que actualizar el alias de rol. No es necesario realizar ningún cambio en el dispositivo. Esta API adopta los siguientes parámetros:

roleAlias

Obligatorio. Una cadena arbitraria que identifica el alias del rol. Sirve como clave principal en el modelo de datos del alias de rol. Contiene 1-128 caracteres y debe incluir únicamente caracteres alfanuméricos y los símbolos =, @ y -. Se permiten los caracteres alfabéticos en mayúsculas y minúsculas.

roleArn

Obligatorio. El ARN del rol al que hace referencia el alias del rol.

credentialDurationSeconds

Opcional. El tiempo (en segundos) que la credencial es válida. El valor mínimo es de 900 segundos (15 minutos). El valor máximo es de 43 200 segundos (12 horas). El valor de predeterminado es de 3,600 segundos (1 hora).

Note

LaAWS IoT CoreEl proveedor de credenciales puede emitir una credencial con una vida útil máxima de 43 200 segundos (12 horas). El hecho de que la credencial sea válida durante un máximo de 12 horas puede ayudar a reducir el número de llamadas al proveedor de credenciales almacenando la credencial durante más tiempo.

La**credentialDurationSeconds**El valor debe ser menor o igual que la duración máxima de la sesión del rol de IAM que hace referencia el alias de rol.

Para obtener más información sobre esta API, consulte [CreateRoleAlias](#).

- Adjunte una política al certificado de dispositivo. La política adjunta al certificado del dispositivo debe conceder permiso al dispositivo para asumir el rol. Para ello, puede conceder permiso para la acción `iot:AssumeRoleWithCertificate` al alias de rol, como en el ejemplo siguiente.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": "iot:AssumeRoleWithCertificate",  
            "Resource": "arn:aws:iot:your_region:your_aws_account_id:rolealias/your_role  
alias"  
        }  
    ]  
}
```

- Realice una solicitud HTTPS al proveedor de credenciales para obtener un token de seguridad. Proporcione la información siguiente:

- Certificado: Dado que se trata de una solicitud HTTP a través de autenticación mutua de TLS, debe proporcionar el certificado y la clave privada al cliente al realizar la solicitud. Utilice el mismo certificado y clave privada utilizados cuando registró su certificado con AWS IoT Core.

Para asegurarse de que un dispositivo se comunica con AWS IoT Core (y no con un servicio que lo suplante), consulte [Autenticación del servidor](#), siga los enlaces para descargar los certificados de entidad de certificación apropiados y, a continuación, cópielos en el dispositivo.

- RoleAlias: El nombre del alias de rol que creó para el proveedor de credenciales.
- ThingName: El nombre de objeto que creó al registrar suAWS IoT Corething. Esto se transfiere como valor del encabezado HTTP `x-amzn-iot-thingname`. Este valor es obligatorio únicamente si utiliza atributos de objetos como variables de política en AWS IoT Core o políticas de IAM.

Note

La `ThingName` que proporciona `enx-amzn-iot-thingname` debe coincidir con el nombre de la AWS IoT Recurso de cosa asignado a un certificado. Si no coincide, se devuelve un error 403.

Ejecute el comando siguiente en la AWS CLI para obtener el punto de enlace del proveedor de credenciales de Cuenta de AWS. Para obtener más información sobre esta API, consulte [DescribeEndpoint](#).

```
aws iot describe-endpoint --endpoint-type iot:CredentialProvider
```

El siguiente objeto de JSON es la salida de ejemplo del comando `describe-endpoint`. Contiene el `endpointAddress` que utiliza para solicitar un token de seguridad.

```
{  
    "endpointAddress": "your_aws_account_specific_prefix.credentials.iot.your  
    region.amazonaws.com"  
}
```

Utilice el punto de enlace para realizar una solicitud HTTPS al proveedor de credenciales para devolver un token de seguridad. El ejemplo de comando siguiente utiliza `curl`, pero puede utilizar cualquier 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
```

Este comando devuelve un objeto de token de seguridad que contiene un `accessKeyId`, una `secretAccessKey`, un `sessionToken` y un vencimiento. El siguiente objeto de JSON es la salida de ejemplo del comando `curl`.

```
{"credentials": {"accessKeyId": "access key", "secretAccessKey": "secret access  
key", "sessionToken": "session token", "expiration": "2018-01-18T09:18:06Z"}}
```

A continuación, puede utilizar el `accessKeyId`, `secretAccessKey`, `sessionToken` valores a los que firmar las solicitudes AWS Services de . Para un end-to-end demostración, véase [Cómo eliminar la necesidad de codificar de forma rígida AWS Credenciales en dispositivos mediante el AWS IoT Proveedor de credenciales](#) Publicación del blog en la AWS Blog de seguridad.

Acceso entre cuentas con IAM

AWS IoT Core le permite habilitar un principal para publicar en un tema o suscribirse a un tema definido en su cuenta de AWS que pertenece al principal. El acceso entre cuentas se configura creando una política de IAM y un rol de IAM y, a continuación, asociando la política al rol.

En primer lugar, cree una política de IAM administrada por el cliente tal como se describe en [Creación de políticas de IAM](#), igual que haría para otros usuarios y certificados en su cuenta de AWS.

Para dispositivos registrados en AWS IoT Core, la siguiente política concede permiso a los dispositivos de conectarse a AWS IoT Core utilizando un ID de cliente que coincide con el nombre de la

cosa del dispositivo y para publicar en el `my/topic/thing-name` where `Nombre de cosa` es el nombre de la cosa del 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 los dispositivos no registrados en el registro de AWS IoT Core, la siguiente política concede permiso a un dispositivo para utilizar el nombre de objeto `client1` registrado en el registro de AWS IoT Core de su cuenta (123456789012) para conectarse a AWS IoT Core y para publicar en un tema específico del ID de cliente cuyo nombre lleva el prefijo `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}"
            ]
        }
    ]
}
```

A continuación, siga los pasos que se indican en [Creación de un rol para delegar permisos a un usuario de IAM](#). Introduzca el ID de cuenta del Cuenta de AWS con el que desea compartir el acceso. A continuación, en el último paso, asocie al rol la política que acaba de crear. Si, más adelante, necesita modificar la AWSID de cuenta al que concede acceso, puede utilizar el siguiente formato de política de confianza:

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

```
{  
    "Effect": "Allow",  
    "Principal": {  
        "AWS": "arn:aws:iam:us-east-1:567890123456:user:MyUser"  
    },  
    "Action": "sts:AssumeRole",  
}  
]  
}
```

Protección de los datos en AWS IoT Core

El [modelo de responsabilidad compartida](#) de AWS se aplica a la protección de datos de AWS IoT Core. Como se describe en este modelo, AWS es responsable de proteger la infraestructura global que ejecuta toda la Nube de AWS. Usted es responsable de mantener el control sobre el contenido alojado en esta infraestructura. Este contenido incluye la configuración de seguridad y las tareas de administración para el que utiliza Servicios de AWS. Para obtener más información sobre la privacidad de los datos, consulte las [Preguntas frecuentes sobre la privacidad de datos](#). Para obtener información sobre la protección de datos en Europa, consulte la publicación de blog [AWS Shared Responsibility Model and GDPR](#) en el Blog de seguridad de AWS.

Con fines de protección de datos, recomendamos proteger las credenciales de Cuenta de AWS y configurar cuentas de usuario individuales con AWS Identity and Access Management (IAM). De esta manera, solo se otorgan a cada usuario los permisos necesarios para cumplir con sus obligaciones laborales. También recomendamos proteger sus datos de las siguientes formas:

- Utilice Multi-Factor Authentication (MFA) con cada cuenta.
- Utilice SSL/TLS para comunicarse con los recursos de AWS. Recomendamos TLS 1.2 o una versión posterior.
- Configure la API y el registro de actividad del usuario con AWS CloudTrail.
- Utilice las soluciones de cifrado de AWS, junto con todos los controles de seguridad predeterminados dentro de los servicios de AWS.
- Utilice avanzados servicios de seguridad administrados, como Amazon Macie, que lo ayuden a detectar y proteger los datos personales almacenados en Amazon S3.
- Si necesita módulos criptográficos validados FIPS 140-2 al acceder a AWS a través de una interfaz de línea de comandos o una API, utilice un punto de enlace de FIPS. Para obtener más información sobre los puntos de enlace de FIPS disponibles, consulte [Estándar de procesamiento de la información federal \(FIPS\) 140-2](#).

Le recomendamos encarecidamente que nunca introduzca información de identificación confidencial, como, por ejemplo, direcciones de email de sus clientes, en etiquetas o en los campos de formato libre, como el campo Name (Nombre). No debe especificar esta información cuando trabaje con AWS IoT u otros servicios de AWS a través de la consola, la API, la AWS CLI o AWS SDK. Los datos que ingresa en etiquetas o campos de formato libre utilizados para los nombres se pueden utilizar para los registros de facturación o diagnóstico. Si proporciona una URL a un servidor externo, recomendamos encarecidamente que no incluya información de credenciales en la URL a fin de validar la solicitud para ese servidor.

Para obtener más información sobre la protección de datos, consulte la entrada de blog relativa al [modelo de responsabilidad compartida de AWS y GDPR](#) en el blog de seguridad de AWS.

Los dispositivos de AWS IoT recopilan datos, realizan alguna manipulación en dichos datos y, a continuación, los envían a otro servicio web. Es posible que decida almacenar algunos datos en su dispositivo durante un breve período de tiempo. Usted es responsable de proporcionar cualquier protección de datos sobre esos datos en reposo. Cuando el dispositivo envía datos a AWS IoT, lo hace mediante una

conexión TLS, como se explica más adelante en esta sección. AWS IoT dispositivos pueden enviar datos a cualquier AWS service. Para obtener más información acerca de la seguridad de datos de cada servicio, consulte la documentación correspondiente a dicho servicio. AWS IoT puede configurar para escribir registros en CloudWatch Registros y registrar AWS IoT llamadas a la API a AWS CloudTrail. Para obtener más información acerca de la seguridad de datos de estos servicios, consulte [Autenticación y control de acceso de Amazon CloudWatch](#) y [Cifrado de CloudTrail Archivos de registro con AWS Claves administradas por SMS](#).

Cifrado de datos en AWS IoT

De forma predeterminada, todos los datos en tránsito y en reposo se cifran. Los datos en tránsito se cifran mediante TLS (p. 382), y los datos en reposo se cifran mediante AWS propias. AWS IoT no admite actualmente administrado por el cliente AWS KMS keys (llaves KMS) desde AWS Key Management Service (AWS KMS); sin embargo, Device Advisor y AWS IoT solo usan una clave propia de AWS para cifrar los datos de los clientes.

Seguridad de transporte en AWS IoT

La AWS IoT agente de mensajes de y el servicio Device Shadow cifran todas las comunicaciones durante el tránsito mediante TLS versión 1.2. TLS se utiliza para garantizar la confidencialidad de los protocolos de aplicación (MQTT, HTTP y WebSocket) que admite AWS IoT. La compatibilidad TLS está disponible en una serie de lenguajes de programación y sistemas operativos. Datos dentro de AWS están encriptados por el AWS service. Para obtener más información acerca del cifrado de datos en otros servicios de AWS, consulte la documentación de seguridad de ese servicio.

Para MQTT, TLS cifra la conexión entre el dispositivo y el agente. AWS IoT utiliza la autenticación de cliente de TLS para identificar los dispositivos. Para HTTP, TLS cifra la conexión entre el dispositivo y el agente. La autenticación se delega en AWS Signature Version 4.

AWS IoT requiere que los dispositivos envíen la extensión Indicación de nombre de servidor (SNI) al protocolo Transport Layer Security (TLS) y proporcionen la dirección completa del punto de enlace en el campo host_name. El campo host_name debe contener el punto de enlace al que está llamando y debe ser:

- El valor de endpointAddress devuelto por `aws iot describe-endpoint --endpoint-type iot:Data-ATS`
o bien
- El valor de domainName devuelto por `aws iot describe-domain-configuration --domain-configuration-name "domain_configuration_name"`

Conexiones intentadas por los dispositivos sin el correcto host_name el valor fallará, y AWS IoT registrará los errores en CloudWatch si el tipo de autenticación es Autenticación personalizada.

AWS IoT es compatible con Session Ticket TLS.

Seguridad de transporte para dispositivos inalámbricos LoRaWAN

Los dispositivos LoRaWAN siguen las prácticas de seguridad descritas en [SEGURIDAD LoRaWAN™: Libro blanco preparado para el LoRa Alliance™ de Gemalto, Actility y Semtech](#).

Para obtener más información acerca de la seguridad del transporte con dispositivos LoRaWAN, consulte [Seguridad de los datos con AWS IoT Core for LoRaWAN \(p. 1209\)](#).

Compatibilidad con el conjunto de cifrado de TLS

AWS IoT admite los conjuntos de cifrado siguientes:

- 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

Cifrado de datos en AWS IoT

La protección de datos se refiere a salvaguardarlos en tránsito (al desplazarse desde y hacia AWS IoT) y en reposo (almacenado en dispositivos o mediante otros servicios de AWS). Todos los datos enviados a AWS IoT se envían a través de una conexión TLS mediante MQTT, HTTPS y WebSocket protocolos, lo que lo hace seguro de forma predeterminada mientras está en tránsito. AWS IoT los dispositivos recopilan datos y luego los envían a otros servicios para su posterior procesamiento. Para obtener más información acerca del cifrado de datos en otros servicios de AWS, consulte la documentación de seguridad de ese servicio.

FreeRTOS proporciona una biblioteca PKCS #11 que resume el almacenamiento de claves, el acceso a objetos criptográficos y la administración de sesiones. Es su responsabilidad utilizar esta biblioteca para cifrar los datos en reposo en sus dispositivos. Para obtener más información, consulte [Biblioteca del estándar de criptografía de clave pública FreeRTOS #11](#).

Asesor de dispositivos

Cifrado en tránsito

Los datos enviados que tienen como origen o destino Device Advisor se cifran en tránsito. Todos los datos enviados hacia y desde el servicio cuando se utilizan las API de Device Advisor se cifran mediante Signature Version 4. Para obtener más información sobre cómo AWS Las solicitudes de API están firmadas, consulte [Firma AWS Solicitud API de](#). Todos los datos enviados desde los dispositivos de prueba al punto final de prueba de Device Advisor se envían a través de una conexión TLS, por lo que es seguro de forma predeterminada en tránsito.

Administración de claves en AWS IoT

Todas las conexiones a AWS IoT se realizan mediante TLS, por lo que no se necesitan claves de cifrado del lado del cliente para la conexión TLS inicial.

Los dispositivos deben autenticarse mediante un certificado X.509 o Amazon Cognito Identity. Puede hacer que AWS IoT genere un certificado para usted, en cuyo caso generará un par de claves públicas/privadas. Si está utilizando la consola de AWS IoT, se le pedirá que descargue el certificado y las claves. Si utiliza [lacreate-keys-and-certificate](#)El comando de la CLI, el certificado y las claves de devuelven el comando de la CLI. Usted es responsable de copiar el certificado y la clave privada en su dispositivo y mantenerlos seguros.

AWS IoT admite actualmente administrado por el cliente AWS KMS keys(llaves KMS) desde AWS Key Management Service(AWS KMS); sin embargo, Device Advisor y AWS IoT solo usan una clave propia de AWS para cifrar los datos de los clientes.

Asesor de dispositivos

Todos los datos enviados a Device Advisor cuando se utiliza el AWS API se cifran en reposo. Device Advisor cifra todos los datos en reposo mediante claves KMS almacenadas y administradas en AWS Key Management Service. Device Advisor cifra los datos mediante claves propiedades de AWS. Para obtener más información acerca de Claves propiedades de AWS, consulte [Claves propiedades de AWS](#).

Identity and Access Management en AWS IoT

AWS Identity and Access Management (IAM) es un Servicio de AWS que ayuda a los administradores a controlar de forma segura el acceso a los recursos de AWS. Los administradores de IAM controlan quién está autenticado (ha iniciado sesión) y autorizado (tiene permisos) para utilizar recursos de AWS IoT. IAM es un Servicio de AWS que se puede utilizar sin cargo adicional.

Temas

- [Público \(p. 384\)](#)
- [Autenticación con identidades de IAM \(p. 385\)](#)
- [Administración de acceso mediante políticas \(p. 387\)](#)
- [Cómo AWS IoT funciona con IAM \(p. 389\)](#)
- [Ejemplos de políticas basadas en identidad de AWS IoT \(p. 408\)](#)
- [Solución de problemas de identidades y accesos en AWS IoT \(p. 411\)](#)

Público

La forma en que utilice AWS Identity and Access Management (IAM) difiere en función del trabajo que realice en AWS IoT.

Usuario de servicio: si utiliza el servicio de AWS IoT para realizar su trabajo, su administrador le proporciona las credenciales y los permisos que necesita. A medida que utilice más características de AWS IoT para realizar su trabajo, es posible que necesite permisos adicionales. Entender cómo se administra el acceso puede ayudarle a solicitar los permisos correctos a su administrador. Si no puede acceder a una característica en AWS IoT, consulte [Solución de problemas de identidades y accesos en AWS IoT \(p. 411\)](#).

Administrador de servicio: si está a cargo de los recursos de AWS IoT en su empresa, probablemente tenga acceso completo a AWS IoT. Su trabajo consiste en determinar qué características y recursos

de AWS IoT deben acceder sus empleados. A continuación, debe enviar solicitudes a su administrador de IAM para cambiar los permisos de los usuarios de su servicio. Revise la información de esta página para conocer los conceptos básicos de IAM. Para obtener más información sobre cómo su empresa puede utilizar IAM con AWS IoT, consulte [Cómo AWS IoT funciona con IAM \(p. 389\)](#).

Administrador de IAM: si es un administrador de IAM, es posible que quiera conocer más detalles sobre cómo escribir políticas para administrar el acceso a AWS IoT. Para consultar ejemplos de políticas basadas en la identidad de AWS IoT que puede utilizar en IAM, consulte [Ejemplos de políticas basadas en identidad de AWS IoT \(p. 408\)](#).

Autenticación con identidades de IAM

En AWS IoT las identidades de pueden ser certificados de dispositivo (X.509), identidades de Amazon Cognito o usuarios o grupos de IAM. En este tema se analizan únicamente identidades de IAM. Para obtener más información acerca de las otras identidades compatibles con AWS IoT, consulte [Autenticación del cliente \(p. 297\)](#).

La autenticación es la manera de iniciar sesión en AWS mediante credenciales de identidad. Para obtener más información acerca de cómo iniciar sesión con la AWS Management Console, consulte [Inicio de sesión en la AWS Management Console como usuario de IAM o usuario raíz](#) en la Guía del usuario de IAM.

Debe estar autenticado (haber iniciado sesión en AWS) como el usuario raíz de la Cuenta de AWS, como un usuario de IAM o asumiendo un rol de IAM. También puede utilizar la autenticación de inicio de sesión único de la empresa o incluso iniciar sesión con Google o Facebook. En estos casos, su administrador habrá configurado previamente la federación de identidad mediante roles de IAM. Cuando obtiene acceso a AWS mediante credenciales de otra empresa, asume un rol indirectamente.

Para iniciar sesión directamente en la [AWS Management Console](#), utilice la contraseña con su dirección de email de usuario raíz o con su nombre de usuario de IAM. Puede acceder a AWS mediante programación utilizando sus claves de acceso de usuario raíz o usuario de IAM. AWS proporciona SDK y herramientas de línea de comandos para firmar criptográficamente su solicitud con sus credenciales. Si no utiliza las herramientas de AWS, debe firmar usted mismo la solicitud. Para ello, utilice Signature Version 4, un protocolo para autenticar solicitudes de API de entrada. Para obtener más información acerca de cómo autenticar solicitudes, consulte [Proceso de firma de Signature Version 4](#) en la Referencia general de AWS.

Independientemente del método de autenticación que utilice, es posible que también deba proporcionar información de seguridad adicional. Por ejemplo, AWS le recomienda el uso de la autenticación multifactor (MFA) para aumentar la seguridad de su cuenta. Para obtener más información, consulte [Uso de la autenticación multifactor \(MFA\) en AWS](#) en la Guía del usuario de IAM.

Cuenta de AWS usuario raíz

Cuando se crea una Cuenta de AWS por primera vez, se comienza con una única identidad de inicio de sesión que tiene acceso completo a todos los recursos y Servicios de AWS de la cuenta. Esta identidad recibe el nombre de usuario raíz de la Cuenta de AWS y se accede a ella iniciando sesión con el email y la contraseña que utilizó para crear la cuenta. Recomendamos encarecidamente que no utilice el usuario raíz en sus tareas cotidianas, ni siquiera en las tareas administrativas. En lugar de ello, es mejor ceñirse a la [práctica recomendada de utilizar el usuario final exclusivamente para crear al primer usuario de IAM](#). A continuación, guarde las credenciales del usuario raíz en un lugar seguro y utilícelas tan solo para algunas tareas de administración de cuentas y servicios.

Usuarios y grupos de IAM

Un [usuario de IAM](#) es una identidad de la Cuenta de AWS que dispone de permisos específicos para una sola persona o aplicación. Un usuario de IAM puede tener credenciales a largo plazo, como un nombre de usuario y una contraseña o un conjunto de claves de acceso. Para obtener información sobre cómo

generar claves de acceso, consulte [Administración de claves de acceso de los usuarios de IAM](#) en la Guía del usuario de IAM. Al generar claves de acceso para un usuario de IAM, asegúrese de ver y guardar de forma segura el par de claves. No puede recuperar la clave de acceso secreta en el futuro. En su lugar, debe generar un nuevo par de claves de acceso.

Un [grupo de IAM](#) es una identidad que especifica un conjunto de usuarios de IAM. No puede iniciar sesión como grupo. Puede usar los grupos para especificar permisos para varios usuarios a la vez. Los grupos facilitan la administración de los permisos de grandes conjuntos de usuarios. Por ejemplo, podría tener un grupo cuyo nombre fuese IAMAdmins y conceder permisos a dicho grupo para administrar los recursos de IAM.

Los usuarios son diferentes de los roles. Un usuario se asocia exclusivamente a una persona o aplicación, pero la intención es que cualquier usuario pueda asumir un rol que necesite. Los usuarios tienen credenciales permanentes a largo plazo y los roles proporcionan credenciales temporales. Para obtener más información, consulte [Cuándo crear un usuario de IAM \(en lugar de un rol\)](#) en la Guía del usuario de IAM.

IAM roles

Un [rol de IAM](#) es una identidad de la Cuenta de AWS que dispone de permisos específicos. Es similar a un usuario de IAM, pero no está asociado a una determinada persona. Puede asumir temporalmente un rol de IAM en la AWS Management Console[cambiando de roles](#). Puede asumir un rol llamando a una operación de la AWS CLI o de la API de AWS, o utilizando una URL personalizada. Para obtener más información acerca de los métodos para el uso de roles, consulte [Uso de roles de IAM](#) en la Guía del usuario de IAM.

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

- Permisos de usuario de IAM temporales: un usuario de IAM puede asumir un rol de IAM para recibir temporalmente permisos distintos que le permitan realizar una tarea concreta.
- Acceso de usuarios federados: en lugar de crear un usuario de IAM, puede utilizar identidades existentes de AWS Directory Service, del directorio de usuarios de su empresa o de un proveedor de identidades web. A estas identidades se les llama usuarios federados. AWS asigna una función a un usuario federado cuando se solicita acceso a través de un [proveedor de identidad](#). Para obtener más información acerca de los usuarios federados, consulte [Usuarios federados y roles](#) en la Guía del usuario de IAM.
- Acceso entre cuentas: puede utilizar un rol de IAM para permitir que alguien (una entidad principal de confianza) de otra cuenta acceda a los recursos de la cuenta. Los roles son la forma principal de conceder acceso entre cuentas. No obstante, con algunos Servicios de AWS se puede adjuntar una política directamente a un recurso (en lugar de utilizar un rol como representante). Para obtener información acerca de la diferencia entre los roles y las políticas basadas en recursos para el acceso entre cuentas, consulte [Cómo los roles de IAM difieren de las políticas basadas en recursos](#) en la Guía del usuario de IAM.
- Acceso entre servicios: algunos Servicios de AWS utilizan características de otros Servicios de AWS. Por ejemplo, cuando realiza una llamada en un servicio, es común que ese servicio ejecute aplicaciones en Amazon EC2 o almacene objetos en Amazon S3. Es posible que un servicio haga esto usando los permisos de la entidad principal, usando un rol de servicio o usando un rol vinculado a servicios.
 - Permisos principales: cuando utiliza un usuario o un rol de IAM para llevar a cabo acciones en AWS, se lo considera una entidad principal. Las políticas conceden permisos a una entidad principal. Cuando utiliza algunos servicios, es posible que realice una acción que desencadene otra acción en un servicio diferente. En este caso, debe tener permisos para realizar ambas acciones. Para ver si una acción requiere acciones dependientes adicionales en una política, consulte [Acciones, recursos y claves de condición de AWS IoT](#) en la Referencia de autorizaciones de servicio.
- Rol de servicio: un rol de servicio es un [rol de IAM](#) que adopta un servicio para realizar acciones en su nombre. Un administrador de IAM puede crear, modificar y eliminar un rol de servicio desde IAM. Para obtener más información, consulte [Creación de un rol para delegar permisos a un Servicio de AWS](#) en la Guía del usuario de IAM.

- Rol vinculado a servicio: un rol vinculado a servicio es un tipo de rol de servicio que está vinculado a un Servicio de AWS. El servicio puede asumir el rol para realizar una acción en su nombre. Los roles vinculados a servicios aparecen en la cuenta de IAM y son propiedad del servicio. Un administrador de IAM puede ver, pero no editar, los permisos de los roles vinculados a servicios.
- Aplicaciones que se ejecutan en Amazon EC2: puede utilizar un rol de IAM que le permita administrar credenciales temporales para las aplicaciones que se ejecutan en una instancia de EC2 y realizar solicitudes a la AWS CLI o a la API de AWS. Es preferible hacerlo de este modo a almacenar claves de acceso en la instancia de EC2. Para asignar un rol de AWS a una instancia de EC2 y ponerla a disposición de todas las aplicaciones, cree un perfil de instancia adjuntado a la instancia. Un perfil de instancia contiene el rol y permite a los programas que se ejecutan en la instancia de EC2 obtener credenciales temporales. Para obtener más información, consulte [Uso de un rol de IAM para conceder permisos a aplicaciones que se ejecutan en instancias Amazon EC2](#) en la Guía del usuario de IAM.

Para obtener información sobre el uso de los roles de IAM, consulte [Cuándo crear un rol de IAM \(en lugar de un usuario\)](#) en la Guía del usuario de IAM.

Administración de acceso mediante políticas

Para controlar el acceso en AWS, se crean políticas y se adjuntan a identidades de IAM o recursos de AWS. Una política es un objeto de AWS que, cuando se asocia a una identidad o un recurso, define sus permisos. Puede iniciar sesión como usuario raíz o usuario de IAM o puede asumir un rol de IAM. Cuando realiza una solicitud, AWS evalúa las políticas relacionadas basadas en identidad o en recursos. Los permisos en las políticas determinan si la solicitud se permite o se deniega. Las mayoría de las políticas se almacenan en AWS como documentos JSON. Para obtener más información sobre la estructura y el contenido de los documentos de política JSON, consulte [Información general de políticas JSON](#) en la Guía del usuario de IAM.

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

Cada entidad de IAM (usuario o rol) comienza sin permisos. En otras palabras, de forma predeterminada, los usuarios no pueden hacer nada, ni siquiera cambiar sus propias contraseñas. Para conceder permiso a un usuario para hacer algo, el administrador debe adjuntarle una política de permisos. O bien el administrador puede agregar al usuario a un grupo que tenga los permisos necesarios. Cuando el administrador concede permisos a un grupo, todos los usuarios de ese grupo obtienen los permisos.

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

Políticas basadas en identidad

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

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

Políticas basadas en recursos

Las políticas basadas en recursos son documentos de política JSON que se adjuntan a un recurso. Ejemplos de políticas basadas en recursos son las políticas de confianza de roles de IAM y las políticas de bucket de Amazon S3. En los servicios que admiten políticas basadas en recursos, los administradores de servicios pueden utilizarlos para controlar el acceso a un recurso específico. Para el recurso al que se adjunta la política, la política define qué acciones puede realizar una entidad principal especificada en ese recurso y en qué condiciones. Debe [especificar una entidad principal](#) en una política basada en recursos. Las entidades principales pueden incluir cuentas, usuarios, roles, usuarios federados o Servicios de AWS.

Las políticas basadas en recursos son políticas insertadas que se encuentran en ese servicio. No se puede utilizar políticas de IAM administradas por AWS en una política basada en recursos.

Listas de control de acceso (ACL)

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

Amazon S3, AWS WAF y Amazon VPC son ejemplos de servicios que admiten las ACL. Para obtener más información sobre las ACL, consulte [Información general de Lista de control de acceso \(ACL\)](#) en la Guía para desarrolladores de Amazon Simple Storage Service.

Otros tipos de políticas

AWS admite otros tipos de políticas adicionales menos frecuentes. Estos tipos de políticas pueden establecer el máximo de permisos que los tipos de políticas más frecuentes le otorgan.

- Límites de permisos: un límite de permisos es una característica avanzada que le permite establecer los permisos máximos que una política basada en identidad puede conceder a una entidad de IAM (usuario o rol de IAM). Puede establecer un límite de permisos para una identidad. Los permisos resultantes son la intersección de las políticas basadas en identidad de la entidad y los límites de sus permisos. Las políticas basadas en recursos que especifiquen el usuario o rol en el campo Principal no estarán restringidas por el límite de permisos. Una denegación explícita en cualquiera de estas políticas anulará el permiso. Para obtener más información sobre los límites de los permisos, consulte [Límites de permisos para las entidades de IAM](#) en la Guía del usuario de IAM.
- Políticas de control de servicio (SCP): las SCP son políticas de JSON que especifican los permisos máximos de una organización o una unidad organizativa en AWS Organizations. AWS Organizations es un servicio que le permite agrupar y administrar de manera centralizada varias Cuentas de AWS que posea su empresa. Si habilita todas las características en una organización, entonces podrá aplicar políticas de control de servicio (SCP) a una o todas sus cuentas. Las SCP limitan los permisos de las entidades de las cuentas miembro, incluido cada usuario raíz de la Cuenta de AWS. Para obtener más información acerca de Organizations y las SCP, consulte [Funcionamiento de las SCP](#) en la Guía del usuario de AWS Organizations.
- Políticas de sesión: las políticas de sesión son políticas avanzadas que se pasan como parámetro cuando se crea una sesión temporal mediante programación para un rol o un usuario federado. Los permisos de la sesión resultantes son la intersección de las políticas basadas en identidad del rol y las políticas de la sesión. Los permisos también pueden proceder de una política basada en recursos. Una denegación explícita en cualquiera de estas políticas anulará el permiso. Para obtener más información, consulte [Políticas de sesión](#) en la Guía del usuario de IAM.

Varios tipos de políticas

Cuando se aplican varios tipos de políticas a una solicitud, los permisos resultantes son más complicados de entender. Para obtener información acerca de cómo AWS decide si permitir o no una solicitud cuando

hay varios tipos de políticas implicados, consulte [Lógica de evaluación de políticas](#) en la Guía del usuario de IAM.

Cómo AWS IoT funciona con IAM

Antes de utilizar IAM para administrar el acceso aAWS IoT, debe saber qué características de IAM están disponibles para usarse conAWS IoT. Para obtener una perspectiva general sobre cómo funcionan AWS IoT y otros servicios de AWS con IAM, consulte [Servicios de AWS que funcionan con IAM](#) en la Guía del usuario de IAM.

Temas

- [Políticas de AWS IoT basadas en identidades \(p. 389\)](#)
- [Políticas de AWS IoT basadas en recursos \(p. 405\)](#)
- [Autorización basada en etiquetas de AWS IoT \(p. 405\)](#)
- [Roles de IAM de AWS IoT \(p. 406\)](#)
- [Políticas administradas de IAM \(p. 406\)](#)

Políticas de AWS IoT basadas en identidades

Con las políticas basadas en identidades de IAM, puede especificar las acciones y recursos permitidos o denegados, así como las condiciones en las que se permiten o deniegan las acciones. AWS IoT admite acciones, recursos y claves de condición específicos. Para obtener información sobre todos los elementos que utiliza en una política JSON, consulte [Referencia de los elementos de las políticas JSON de IAM](#) en la Guía del usuario de IAM.

Acciones

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

El elemento `Action` de una política JSON describe las acciones que puede utilizar para permitir o denegar el acceso en una política. Las acciones de la política generalmente tienen el mismo nombre que la operación de API de AWS asociada. Hay algunas excepciones, como acciones de solo permiso que no tienen una operación de API coincidente. También hay algunas operaciones que requieren varias acciones en una política. Estas acciones adicionales se denominan acciones dependientes.

Incluya acciones en una política para conceder permisos y así llevar a cabo la operación asociada.

En la siguiente tabla se enumeran las acciones de IoT de IAM, lasAWS IoTAPI y el recurso que manipula la acción.

Acciones de política	API de AWS IoT	Recursos
iot:AcceptCertificateTransfer	iot:AcceptCertificateTransfer	<code>arn:aws:iot:<region>:<account-id>:cert/<cert-id></code> Note La Cuenta de AWS especificada en el ARN debe ser la cuenta a la que se transfiere el certificado.
iot:AddThingToThingGroup	iot:AddThingToThingGroup	<code>arn:aws:iot:<region>:<account-id>:thinggroup/<thing-group-name></code> <code>arn:aws:iot:<region>:<account-id>:thing/<thing-name></code>
iot:AssociateTargetsWithThing	iot:AssociateTargetsWithThing	Any

Acciones de política	API de AWS IoT	Recursos
iot:AttachPolicy	AttachPolicy	arn:aws:iot: region:account-id:thinggroup/thing-group-name o bien 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 La Cuenta de AWS especificada en el ARN debe ser la cuenta a la que se transfiere el certificado.
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	Ninguno
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
IoT: crear plantilla de trabajo	CreateJobTemplate	arn:aws:iot: region:account-id:job/job-id arn:aws:iot: region:account-id:jobtemplate/job-template-id
iot>CreateKeysAndCertificate	CreateKeysAndCertificate	

Acciones de política	API de AWS IoT	Recursos
iot:CreatePolicy	CreatePolicy	arn:aws:iot: <i>region:account-id:policy/policy-name</i>
iot:CreatePolicyVersion	CreatePolicyVersion	arn:aws:iot: <i>region:account-id:policy/policy-name</i> Note Deben ser una AWS IoT política, no una política de IAM.
iot:CreateRoleAlias	CreateRoleAlias	(parámetro: roleAlias) arn:aws:iot: <i>region:account-id:rolealias/role-alias-name</i>
iot:CreateSecurityProfile	CreateSecurityProfile	arn:aws:iot: <i>region:account-id:securityprofile/security-profile-name</i> arn:aws:iot: <i>region:account-id:dimension/dimension-name</i>
iot:CreateThing	CreateThing	arn:aws:iot: <i>region:account-id:thing/thing-name</i>
iot:CreateThingGroup	CreateThingGroup	arn:aws:iot: <i>region:account-id:thinggroup/thing-group-name</i> para el grupo que se está creando y para el grupo principal, si se utiliza
iot:CreateThingType	CreateThingType	arn:aws:iot: <i>region:account-id:thingtype/thing-type-name</i>
iot:CreateTopicRule	CreateTopicRule	arn:aws:iot: <i>region:account-id:rule/rule-name</i>
iot:DeleteAuthorizer	DeleteAuthorizer	arn:aws:iot: <i>region:account-id:authorizer/authorizer-name</i>
iot:DeleteCACertificate	DeleteCACertificate	arn:aws:iot: <i>region:account-id:cacert/cert-id</i>
iot:DeleteCertificate	DeleteCertificate	arn:aws:iot: <i>region:account-id:cert/cert-id</i>
iot:DeleteDimension	DeleteDimension	arn:aws:iot: <i>region:account-id:dimension/dimension-name</i>
iot:DeleteJob	DeleteJob	arn:aws:iot: <i>region:account-id:job/job-id</i>
IoT: eliminar plantilla de trabajo	DeleteJobTemplate	arn:aws:iot: <i>region:account-id:job/job-template-id</i>
iot:DeleteJobExecution	DeleteJobExecution	arn:aws:iot: <i>region:account-id:job/job-id</i> arn:aws:iot: <i>region:account-id:thing/thing-name</i>
iot:DeletePolicy	DeletePolicy	arn:aws:iot: <i>region:account-id:policy/policy-name</i>

Acciones de política	API de AWS IoT	Recursos
iot:DeletePolicyVersion	DeletePolicyVersion	arn:aws:iot: <i>region:account-id:policy/policy-name</i>
iot:DeleteRegistrationCode	DeleteRegistrationCode	
iot:DeleteRoleAlias	DeleteRoleAlias	arn:aws:iot: <i>region:account-id:rolealias/rolealias-name</i>
iot:DeleteSecurityProfile	DeleteSecurityProfile	arn:aws:iot: <i>region:account-id:securityprofile/security-profile-name</i> arn:aws:iot: <i>region:account-id:dimension/dimension-name</i>
iot:DeleteThing	DeleteThing	arn:aws:iot: <i>region:account-id:thing/thing-name</i>
iot:DeleteThingGroup	DeleteThingGroup	arn:aws:iot: <i>region:account-id:thinggroup/thing-group-name</i>
iot:DeleteThingType	DeleteThingType	arn:aws:iot: <i>region:account-id:thingtype/thing-type-name</i>
iot:DeleteTopicRule	DeleteTopicRule	arn:aws:iot: <i>region:account-id:rule/rule-name</i>
iot:DeleteV2LoggingLevel	DeleteV2LoggingLevel	arn:aws:iot: <i>region:account-id:thinggroup/thing-group-name</i>
iot:DeprecateThingType	DeprecateThingType	arn:aws:iot: <i>region:account-id:thingtype/thing-type-name</i>
iot:DescribeAuthorizer	DescribeAuthorizer	arn:aws:iot: <i>region:account-id:authorizer/authorizer-function-name</i> (parámetro: authorizerName) ninguno
iot:DescribeCACertificate	DescribeCACertificate	arn:aws:iot: <i>region:account-id:cacert/cert-id</i>
iot:DescribeCertificate	DescribeCertificate	arn:aws:iot: <i>region:account-id:cert/cert-id</i>
iot:DescribeDefaultAuthorizer	DescribeDefaultAuthorizer	Ninguno
iot:DescribeEndpoint	DescribeEndpoint	*
iot:DescribeEventConfiguration	DescribeEventConfigurations	
iot:DescribeIndex	DescribeIndex	arn:aws:iot: <i>region:account-id:index/index-name</i>
iot:DescribeJob	DescribeJob	arn:aws:iot: <i>region:account-id:job/job-id</i>
iot:DescribeJobExecution	DescribeJobExecution	Ninguno
IoT: descripción de la plantilla de trabajo	Describir plantilla de trabajo	arn:aws:iot: <i>region:account-id:job/job-template-id</i>
iot:DescribeRoleAlias	DescribeRoleAlias	arn:aws:iot: <i>region:account-id:rolealias/rolealias-name</i>

Acciones de política	API de AWS IoT	Recursos
iot:DescribeThing	DescribeThing	arn:aws:iot: <i>region:account-id:thing/thing-name</i>
iot:DescribeThingGroup	DescribeThingGroup	arn:aws:iot: <i>region:account-id:thinggroup/thing-group-name</i>
iot:DescribeThingRegistrationTask	DescribeThingRegistrationTask	
iot:DescribeThingType	DescribeThingType	arn:aws:iot: <i>region:account-id:thingtype/thing-type-name</i>
iot:DetachPolicy	DetachPolicy	arn:aws:iot: <i>region:account-id:cert/cert-id</i> o bien arn:aws:iot: <i>region:account-id:thinggroup/thing-group-name</i>
iot:DetachPrincipalPolicy	DetachPrincipalPolicy	arn:aws:iot: <i>region:account-id:cert/cert-id</i>
iot:DetachSecurityProfile	DetachSecurityProfile	arn:aws:iot: <i>region:account-id:securityprofile/security-profile-name</i> arn:aws:iot: <i>region:account-id:dimension/dimension-name</i>
iot:DetachThingPrincipal	DetachThingPrincipal	arn:aws:iot: <i>region:account-id:cert/cert-id</i>
iot:DisableTopicRule	DisableTopicRule	arn:aws:iot: <i>region:account-id:rule/rule-name</i>
iot:EnableTopicRule	EnableTopicRule	arn:aws:iot: <i>region:account-id:rule/rule-name</i>
iot:GetEffectivePolicies	GetEffectivePolicies	arn:aws:iot: <i>region:account-id:cert/cert-id</i>
iot:GetIndexingConfig	GetIndexingConfiguration	Ninguno
iot:GetJobDocument	GetJobDocument	arn:aws:iot: <i>region:account-id:job/job-id</i>
iot:GetLoggingOptions	GetLoggingOptions	*
iot:GetPolicy	GetPolicy	arn:aws:iot: <i>region:account-id:policy/policy-name</i>
iot:GetPolicyVersion	GetPolicyVersion	arn:aws:iot: <i>region:account-id:policy/policy-name</i>
iot:GetRegistrationCode	GetRegistrationCode	*
iot:GetTopicRule	GetTopicRule	arn:aws:iot: <i>region:account-id:rule/rule-name</i>
iot>ListAttachedPolicies	ListAttachedPolicies	arn:aws:iot: <i>region:account-id:thinggroup/thing-group-name</i> o bien arn:aws:iot: <i>region:account-id:cert/cert-id</i>
iot>ListAuthorizers	ListAuthorizers	Ninguno
iot>ListCACertificates	ListCACertificates	*

Acciones de política	API de AWS IoT	Recursos
iot:ListCertificates	ListCertificates	*
iot:ListCertificatesByCA	ListCertificatesByCA	*
iot:ListIndices	ListIndices	Ninguno
iot:ListJobExecutions	ListJobExecutions	Ninguno
iot:ListJobExecutionsForThing	ListJobExecutionsForThing	Ninguno
iot:ListJobs	ListJobs	arn:aws:iot: <i>region:account-id:thinggroup/thing-group-name</i> si thingGroupName parámetro utilizado
IoT: Plantillas de trabajo de lista	ListJobs	Ninguno
iot:ListOutgoingCertificates	ListOutgoingCertificates	
iot:ListPolicies	ListPolicies	*
iot:ListPolicyPrincipals	ListPolicyPrincipals	arn:aws:iot: <i>region:account-id:policy/policy-name</i>
iot:ListPolicyVersions	ListPolicyVersions	arn:aws:iot: <i>region:account-id:policy/policy-name</i>
iot:ListPrincipalPolicies	ListPrincipalPolicies	arn:aws:iot: <i>region:account-id:cert/cert-id</i>
iot:ListPrincipalThings	ListPrincipalThings	arn:aws:iot: <i>region:account-id:cert/cert-id</i>
iot:ListRoleAliases	ListRoleAliases	Ninguno
iot:ListTargetsForPolicy	ListTargetsForPolicy	arn:aws:iot: <i>region:account-id:policy/policy-name</i>
iot:ListThingGroups	ListThingGroups	Ninguno
iot:ListThingGroupsForThing	ListThingGroupsForThing	arn:aws:iot: <i>region:account-id:thing/thing-name</i>
iot:ListThingPrincipals	ListThingPrincipals	arn:aws:iot: <i>region:account-id:thing/thing-name</i>
iot:ListThingRegistrations	ListThingRegistrations	Ninguno
iot:ListThingRegistrations	ListThingRegistrations	Ninguno
iot:ListThingTypes	ListThingTypes	*
iot:ListThings	ListThings	*
iot:ListThingsInThingGroup	ListThingsInThingGroup	arn:aws:iot: <i>region:account-id:thinggroup/thing-group-name</i>
iot:ListTopicRules	ListTopicRules	*
iot:ListV2LoggingLevels	ListV2LoggingLevels	Ninguno
iot:RegisterCACertificate	RegisterCACertificate	*

Acciones de política	API de AWS IoT	Recursos
iot:RegisterCertificate	RegisterCertificate	*
iot:RegisterThing	RegisterThing	Ninguno
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	Ninguno
iot:StopThingRegistrationTask	StopThingRegistrationTask	Ninguno
iot:TestAuthorization	TestAuthorization	arn:aws:iot: region:account-id:cert/cert-id
iot:TestInvokeAuthorizer	TestInvokeAuthorizer	Ninguno
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:UpdateEventConfig	UpdateEventConfig	Ninguno
iot:UpdateIndexingConfig	UpdateIndexingConfig	Ninguno
iot:UpdateRoleAlias	UpdateRoleAlias	arn:aws:iot: region:account-id:rolealias/rolealias-name

Acciones de política	API de AWS IoT	Recursos
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:UpdateThingGroups	UpdateThingGroupsForThings	iot: region:account-id:thing/thing-name

Acciones de política de AWS IoT Use el siguiente prefijo antes de la acción: `iot:`. Por ejemplo, para conceder a alguien permiso para obtener una lista de todas las cosas de IoT registradas en su cuenta de AWS con `ListThings` API, incluya el `iot:ListThings` acción en su política. Las instrucciones de política deben incluir un elemento `Action` o `NotAction`. AWS IoT define su propio conjunto de acciones que describen las tareas que se pueden realizar con este servicio.

Para especificar varias acciones en una única instrucción, sepárelas con comas del siguiente modo:

```
"Action": [
    "ec2:action1",
    "ec2:action2"
```

Puede utilizar caracteres comodín para especificar varias acciones (*). Por ejemplo, para especificar todas las acciones que comienzan con la palabra `Describe`, incluya la siguiente acción:

```
"Action": "iot:Describe*"
```

Para ver una lista de las acciones de AWS IoT, consulte [Acciones definidas por AWS IoT](#) en la Guía del usuario de IAM.

Acciones de Device Advisor

En la siguiente tabla se enumeran las acciones de IAM IoT Device Advisor, AWS IoT API de Device Advisor y el recurso que manipula la acción.

Acciones de política	API de AWS IoT	Recursos
Asesor de dispositivos IoT: Crear definición de Suite	Crear definición de suite	Ninguno
Asesor de dispositivos IoT: Eliminar definición de suite	Eliminar definición de suite	arn:aws:iotdeviceadvisor: region:account-id:suitedefinition/suite-definition-id
Asesor de dispositivos IoT:	Definición de Get Suite	arn:aws:iotdeviceadvisor: region:account-id:suitedefinition/suite-definition-id

Acciones de política	API de AWS IoT	Recursos
Obtener definición de Suite		
Asesor de dispositivos IoT: Get Suite Run	Get Suite Run	arn:aws:iotdeviceadvisor: <i>region:account-id:suitedefinition/suite-run-id</i>
Asesor de dispositivos IoT: Informe de ejecución de GetSuit	Informe de ejecución de GetSuit	arn:aws:iotdeviceadvisor: <i>region:account-id:suiterun/suite-definition-id/suite-run-id</i>
Asesor de dispositivos IoT: definiciones de la suite de listas	Definiciones de ListSuite	Ninguno
Asesor de dispositivos IoT: lista de ejecuciones de suite	Lista Suit Runs	arn:aws:iotdeviceadvisor: <i>region:account-id:suitedefinition/suite-definition-id</i>
IoT Device Advisor: lista de etiquetas para recursos	ListTagsForResource	arn:aws:iotdeviceadvisor: <i>region:account-id:suitedefinition/suite-definition-id</i> arn:aws:iotdeviceadvisor: <i>region:account-id:suiterun/suite-definition-id/suite-run-id</i>
Asesor de dispositivos IoT: Iniciar la ejecución de Suit	Iniciar Suit Run	arn:aws:iotdeviceadvisor: <i>region:account-id:suitedefinition/suite-definition-id</i>
Asesor de dispositivos IoT: recurso de etiquetas	TagResource	arn:aws:iotdeviceadvisor: <i>region:account-id:suitedefinition/suite-definition-id</i> arn:aws:iotdeviceadvisor: <i>region:account-id:suiterun/suite-definition-id/suite-run-id</i>
Asesor de dispositivos IoT: recurso Untag	UntagResource	arn:aws:iotdeviceadvisor: <i>region:account-id:suitedefinition/suite-definition-id</i> arn:aws:iotdeviceadvisor: <i>region:account-id:suiterun/suite-definition-id/suite-run-id</i>
Asesor de dispositivos IoT: actualización de la definición de la suite	Actualización de la definición de la suite	arn:aws:iotdeviceadvisor: <i>region:account-id:suitedefinition/suite-definition-id</i>
Asesor de dispositivos IoT: Stop Suite Run	Stop Suite Run	arn:aws:iotdeviceadvisor: <i>region:account-id:suiterun/suite-definition-id/suite-run-id</i>

Acciones de política de AWS IoT Device Advisor utilice el siguiente prefijo antes de la acción: `iotdeviceadvisor:`. Por ejemplo, para conceder a alguien permiso para obtener una lista de todas las definiciones de suite registradas en su cuenta de AWS con ListSuiteDefinitions API, incluye `eliotdeviceadvisor>ListSuiteDefinitions` acción en su política.

Recursos

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

El elemento `Resource` de la política JSON especifica el objeto u objetos a los que se aplica la acción. Las instrucciones deben contener un elemento `Resource` o `NotResource`. Como práctica recomendada, especifique un recurso utilizando el [Nombre de recurso de Amazon \(ARN\)](#). Puede hacerlo para acciones que admitan un tipo de recurso específico, conocido como permisos de nivel de recurso.

Para las acciones que no admiten permisos de nivel de recurso, como las operaciones de descripción, utilice un carácter comodín (*) para indicar que la instrucción se aplica a todos los recursos.

```
"Resource": "*"
```

Recursos de AWS IoT

Acciones de política	API de AWS IoT	Recursos
iot:AcceptCertificateTransfer	AcceptCertificateTransfer	<code>arn:aws:iot:region:account-id:cert/cert-id</code> Note La cuenta de AWS especificada en el ARN debe ser la cuenta a la que se transfiere el certificado.
iot:AddThingToThingGroup	AddThingToThingGroup	<code>arn:aws:iot:region:account-id:thinggroup/thing-group-name</code> <code>arn:aws:iot:region:account-id:thing/thing-name</code>
iot:AssociateTargetsWithJob	AssociateTargetsWithJob	None
iot:AttachPolicy	AttachPolicy	<code>arn:aws:iot:region:account-id:thinggroup/thing-group-name</code> o bien <code>arn:aws:iot:region:account-id:cert/cert-id</code>
iot:AttachPrincipalPolicy	AttachPrincipalPolicy	<code>arn:aws:iot:region:account-id:cert/cert-id</code>
iot:AttachThingPrincipal	AttachThingPrincipal	<code>arn:aws:iot:region:account-id:cert/cert-id</code>
iot:CancelCertificateTransfer	CancelCertificateTransfer	<code>arn:aws:iot:region:account-id:cert/cert-id</code> Note La cuenta de AWS especificada en el ARN debe ser la cuenta a la que se transfiere el certificado.
iot:CancelJob	CancelJob	<code>arn:aws:iot:region:account-id:job/job-id</code>
iot:CancelJobExecution	CancelJobExecution	<code>arn:aws:iot:region:account-id:job/job-id</code>

Acciones de política	API de AWS IoT	Recursos
		<code>arn:aws:iot:<region>:<account-id>:thing/<thing-name></code>
iot:ClearDefaultAuthorizer	ClearDefaultAuthorizer	Ninguno
iot:CreateAuthorizer	CreateAuthorizer	<code>arn:aws:iot:<region>:<account-id>:authorizer/<authorizer-function-name></code>
iot:CreateCertificateFromCsr	CreatesCertificateFromCsr	
iot:CreateJob	CreateJob	<code>arn:aws:iot:<region>:<account-id>:job/<job-id></code> <code>arn:aws:iot:<region>:<account-id>:thinggroup/<thing-group-name></code> <code>arn:aws:iot:<region>:<account-id>:thing/<thing-name></code> <code>arn:aws:iot:<region>:<account-id>:jobtemplate/<job-template-id></code>
IoT: crear plantilla de trabajo	CreateJobTemplate	<code>arn:aws:iot:<region>:<account-id>:job/<job-id></code> <code>arn:aws:iot:<region>:<account-id>:jobtemplate/<job-template-id></code>
iot:CreateKeysAndCertificate	CreateKeysAndCertificate	
iot:CreatePolicy	CreatePolicy	<code>arn:aws:iot:<region>:<account-id>:policy/<policy-name></code>
CreatePolicyVersion	iot:CreatePolicyVersion	<code>arn:aws:iot:<region>:<account-id>:policy/<policy-name></code> <p>Note</p> <p>Deben ser unAWS IoT política, no una política de IAM.</p>
iot:CreateRoleAlias	CreateRoleAlias	(parámetro: roleAlias) <code>arn:aws:iot:<region>:<account-id>:rolealias/<role-alias-name></code>
iot:CreateThing	CreateThing	<code>arn:aws:iot:<region>:<account-id>:thing/<thing-name></code>
iot:CreateThingGroup	CreateThingGroup	<code>arn:aws:iot:<region>:<account-id>:thinggroup/<thing-group-name></code> para el grupo que se está creando y para el grupo principal, si se utiliza
iot:CreateThingType	CreateThingType	<code>arn:aws:iot:<region>:<account-id>:thingtype/<thing-type-name></code>
iot:CreateTopicRule	CreateTopicRule	<code>arn:aws:iot:<region>:<account-id>:rule/<rule-name></code>
iot:DeleteAuthorizer	DeleteAuthorizer	<code>arn:aws:iot:<region>:<account-id>:authorizer/<authorizer-name></code>
iot:DeleteCACertificate	DeleteCACertificate	<code>arn:aws:iot:<region>:<account-id>:cacert/<cert-id></code>

Acciones de política	API de AWS IoT	Recursos
iot:DeleteCertificate	DeleteCertificate	arn:aws:iot: <i>region:account-id:cert/cert-id</i>
iot:DeleteJob	DeleteJob	arn:aws:iot: <i>region:account-id:job/job-id</i>
iot:DeleteJobExecution	DeleteJobExecution	arn:aws:iot: <i>region:account-id:job/job-id</i> arn:aws:iot: <i>region:account-id:thing/thing-name</i>
IoT: eliminar plantilla de trabajo	DeleteJobTemplate	arn:aws:iot: <i>region:account-id:jobtemplate/job-template-id</i>
iot:DeletePolicy	DeletePolicy	arn:aws:iot: <i>region:account-id:policy/policy-name</i>
iot:DeletePolicyVersion	DeletePolicyVersion	arn:aws:iot: <i>region:account-id:policy/policy-name</i>
iot:DeleteRegistrationCode	DeleteRegistrationCode	
iot:DeleteRoleAlias	DeleteRoleAlias	arn:aws:iot: <i>region:account-id:rolealias/role-alias-name</i>
iot:DeleteThing	DeleteThing	arn:aws:iot: <i>region:account-id:thing/thing-name</i>
iot:DeleteThingGroup	DeleteThingGroup	arn:aws:iot: <i>region:account-id:thinggroup/thing-group-name</i>
iot:DeleteThingType	DeleteThingType	arn:aws:iot: <i>region:account-id:thingtype/thing-type-name</i>
iot:DeleteTopicRule	DeleteTopicRule	arn:aws:iot: <i>region:account-id:rule/rule-name</i>
iot:DeleteV2LoggingLevel	DeleteV2LoggingLevel	arn:aws:iot: <i>region:account-id:thinggroup/thing-group-name</i>
iot:DeprecateThingType	DeprecateThingType	arn:aws:iot: <i>region:account-id:thingtype/thing-type-name</i>
iot:DescribeAuthorizer	DescribeAuthorizer	arn:aws:iot: <i>region:account-id:authorizer/authorizer-function-name</i> (parámetro: authorizerName) ninguno
iot:DescribeCACertificate	DescribeCACertificate	arn:aws:iot: <i>region:account-id:cacert/cert-id</i>
iot:DescribeCertificate	DescribeCertificate	arn:aws:iot: <i>region:account-id:cert/cert-id</i>
iot:DescribeDefaultAuthorizer	DescribeDefaultAuthorizer	ninguno
iot:DescribeEndpoint	DescribeEndpoint	*
iot:DescribeEventConfiguration	DescribeEventConfiguration	
iot:DescribeIndex	DescribeIndex	arn:aws:iot: <i>region:account-id:index/index-name</i>
iot:DescribeJob	DescribeJob	arn:aws:iot: <i>region:account-id:job/job-id</i>
iot:DescribeJobExecution	DescribeJobExecution	ninguno

Acciones de política	API de AWS IoT	Recursos
IoT: descripción de la plantilla de trabajo	Describir plantilla de trabajo	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	Ninguno
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 o bien arn:aws:iot: region:account-id:thinggroup/thing-group-name
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:GetIndexingConfig	GetIndexingConfiguration	Ninguno
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 o bien arn:aws:iot: region:account-id:cert/cert-id
iot>ListAuthorizers	ListAuthorizers	Ninguno

Acciones de política	API de AWS IoT	Recursos
iot:ListCACertificates	ListCACertificates	*
iot:ListCertificates	ListCertificates	*
iot:ListCertificatesByCA	ListCertificatesByCA	*
iot:ListIndices	ListIndices	Ninguno
iot:ListJobExecutions	ListJobExecutions	None
iot:ListJobExecutions	ListJobExecutions	None
iot:ListJobs	ListJobs	arn:aws:iot: <i>region:account-id:thinggroup/thing-group-name</i> si thingGroupName parámetro utilizado
IoT: Plantillas de trabajo de lista	Plantillas de trabajo de lista	Ninguno
iot:ListOutgoingCertificates	ListOutgoingCertificates	*
iot:ListPolicies	ListPolicies	*
iot:ListPolicyPrincipals	ListPolicyPrincipals	arn:aws:iot: <i>region:account-id:policy/policy-name</i>
iot:ListPolicyVersions	ListPolicyVersions	arn:aws:iot: <i>region:account-id:policy/policy-name</i>
iot:ListPrincipalPolicies	ListPrincipalPolicies	arn:aws:iot: <i>region:account-id:cert/cert-id</i>
iot:ListPrincipalThings	ListPrincipalThings	arn:aws:iot: <i>region:account-id:cert/cert-id</i>
iot:ListRoleAliases	ListRoleAliases	Ninguno
iot:ListTargetsForPolicy	ListTargetsForPolicy	arn:aws:iot: <i>region:account-id:policy/policy-name</i>
iot:ListThingGroups	ListThingGroups	Ninguno
iot:ListThingGroupsForThing	ListThingGroupsForThing	arn:aws:iot: <i>region:account-id:thing/thing-name</i>
iot:ListThingPrincipals	ListThingPrincipals	arn:aws:iot: <i>region:account-id:thing/thing-name</i>
iot:ListThingRegistrations	ListThingRegistrations	arn:aws:iot: <i>region:account-id:task/task-report</i>
iot:ListThingRegistrations	ListThingRegistrations	Ninguno
iot:ListThingTypes	ListThingTypes	*
iot:ListThings	ListThings	*
iot:ListThingsInThingGroup	ListThingsInThingGroup	arn:aws:iot: <i>region:account-id:thinggroup/thing-group-name</i>
iot:ListTopicRules	ListTopicRules	*
iot:ListV2LoggingLevels	ListV2LoggingLevels	Ninguno

Acciones de política	API de AWS IoT	Recursos
iot:RegisterCACertificate*	RegisterCACertificate*	
iot:RegisterCertificate	RegisterCertificate	*
iot:RegisterThing	RegisterThing	Ninguno
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:StartThingRegistration	StartThingRegistration	Ninguno
iot:StopThingRegistration	StopThingRegistration	Ninguno
iot:TestAuthorization	TestAuthorization	arn:aws:iot: region:account-id:cert/cert-id
iot:TestInvokeAuthorizer	TestInvokeAuthorizer	Ninguno
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:UpdateEventConfiguration	UpdateEventConfiguration	Ninguno
iot:UpdateIndexingConfiguration	UpdateIndexingConfiguration	Ninguno
iot:UpdateRoleAlias	UpdateRoleAlias	arn:aws:iot: region:account-id:rolealias/rolealias-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

Acciones de política	API de AWS IoT	Recursos
iot:UpdateThingGroups	iot:UpdateThingGroupsForThings	iot: region:account-id:thing/thing-name

Para obtener más información acerca del formato de los ARN, consulte [Nombres de recursos de Amazon \(ARN\) y espacios de nombres de servicios de AWS](#).

Algunas acciones de AWS IoT, como las empleadas para la creación de recursos, no se pueden llevar a cabo en un recurso específico. En dichos casos, debe utilizar el carácter comodín (*).

```
"Resource": "*"
```

Para ver una lista de tipos de recursos y sus ARN, consulte [Recursos definidos por AWS IoT en la IAM User Guide](#). Para obtener información sobre las acciones con las que puede especificar el ARN de cada recursos, consulte [Acciones definidas por AWS IoT](#).

Recursos de Device Advisor

Para definir restricciones de nivel de recursos para AWS IoT las directivas de IAM de Device Advisor utilizan los siguientes formatos ARN de recursos para definiciones de suites y ejecuciones de suites.

Formato ARN de recurso de definición de suite

```
arn:aws:iotdeviceadvisor:region:account-id:suitedefinition/suite-definition-id
```

Formato ARN de recurso de ejecución de suite

```
arn:aws:iotdeviceadvisor:region:account-id:suiterun/suite-definition-id/suite-run-id
```

Claves de condición

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

El elemento `Condition` (o bloque de `Condition`) permite especificar condiciones en las que entra en vigor una instrucción. El elemento `Condition` es opcional. Puede crear expresiones condicionales que utilicen [operadores de condición](#), tales como igual o menor que, para que la condición de la política coincida con los valores de la solicitud.

Si especifica varios elementos de `Condition` en una instrucción o varias claves en un único elemento de `Condition`, AWS las evalúa mediante una operación AND lógica. Si especifica varios valores para una única clave de condición, AWS evalúa la condición con una operación lógica OR. Se deben cumplir todas las condiciones antes de que se concedan los permisos de la instrucción.

También puede utilizar variables de marcador de posición al especificar condiciones. Por ejemplo, puede conceder un permiso de usuario de IAM para acceder a un recurso solo si está etiquetado con su nombre de usuario de IAM. Para obtener más información, consulte [Elementos de la política de IAM: variables y etiquetas](#) en la Guía del usuario de IAM.

AWS admite claves de condición globales y claves de condición específicas del servicio. Para ver todas las claves de condición globales de AWS, consulte [Claves de contexto de condición globales de AWS](#) en la Guía del usuario de IAM.

AWS IoT define su propio conjunto de claves de condición y también admite el uso de algunas claves de condición globales. Para ver todas las claves de condición globales de AWS, consulte [Claves de contexto de condición globales de AWS](#) en la Guía del usuario de IAM.

Claves de condición de AWS IoT

Claves de condición de AWS IoT	Descripción	Tipo
<code>aws:RequestTag/\${tag-key}</code>	Una clave de etiqueta que está presente en la solicitud que el usuario realiza a AWS IoT.	Cadena
<code>aws:ResourceTag/\${tag-key}</code>	El componente de clave de etiqueta de una etiqueta asociada a un recurso de AWS IoT.	Cadena
<code>aws:TagKeys</code>	La lista de todos los nombres de clave de etiqueta asociados con el recurso de la solicitud.	Cadena

Para ver una lista de AWS IoT Claves de condición, consulte [Claves de condición de AWS IoT](#) en la IAM User Guide. Para obtener más información acerca de las acciones y los recursos con los que puede utilizar una clave de condición, consulte [Acciones definidas por AWS IoT](#).

Ejemplos

Para ver ejemplos de políticas basadas en identidad de AWS IoT, consulte [Ejemplos de políticas basadas en identidad de AWS IoT \(p. 408\)](#).

Políticas de AWS IoT basadas en recursos

Las políticas basadas en recursos son documentos de política JSON que especifican qué acciones puede realizar una entidad principal especificada en el recurso de AWS IoT y en qué condiciones.

AWS IoT no admite las políticas basadas en recursos de IAM. Sin embargo, admite AWS IoT Políticas basadas en recursos de . Para obtener más información, consulte [Políticas de AWS IoT Core \(p. 333\)](#).

Autorización basada en etiquetas de AWS IoT

Puede adjuntar etiquetas a los recursos de AWS IoT o transferirlas en una solicitud a AWS IoT. Para controlar el acceso en función de etiquetas, debe proporcionar información de las etiquetas en el [elemento de condición](#) de una política utilizando las claves de condición `iot:ResourceTag/key-name`, `aws:RequestTag/key-name` o `aws:TagKeys`. Para obtener más información, consulte [Uso de etiquetas con políticas de IAM \(p. 289\)](#). Para obtener más información acerca del etiquetado de recursos de AWS IoT, consulte [Etiquetado de los recursos de AWS IoT \(p. 288\)](#).

Para consultar un ejemplo de política basada en la identidad para limitar el acceso a un recurso en función de las etiquetas de ese recurso, consulte [Visualización de recursos de AWS IoT basados en etiquetas \(p. 410\)](#).

Roles de IAM de AWS IoT

Un [rol de IAM](#) es una entidad de la Cuenta de AWS que dispone de permisos específicos.

Uso de credenciales temporales con AWS IoT

Puede utilizar credenciales temporales para iniciar sesión con federación, asumir un rol de IAM o asumir un rol de acceso entre cuentas. Las credenciales de seguridad temporales se obtienen mediante una llamada a operaciones de la API de AWS STS, como [AssumeRole](#) o [GetFederationToken](#).

AWS IoT admite el uso de credenciales temporales.

Roles vinculados a servicios

Los [roles vinculados a servicios](#) permiten a los servicios de AWS obtener acceso a los recursos de otros servicios para completar una acción en su nombre. Los roles vinculados a servicios aparecen en la cuenta de IAM y son propiedad del servicio. Un administrador de IAM puede ver, pero no editar, los permisos de los roles vinculados a servicios.

AWS IoT no admite roles vinculados a servicios

Roles de servicio

Esta característica permite que un servicio asuma un [rol de servicio](#) en su nombre. Este rol permite que el servicio obtenga acceso a los recursos de otros servicios para completar una acción en su nombre. Los roles de servicio aparecen en su cuenta de IAM y son propiedad de la cuenta. Esto significa que un administrador de IAM puede cambiar los permisos de este rol. Sin embargo, hacerlo podría deteriorar la funcionalidad del servicio.

Políticas administradas de IAM

AWS IoT funciona con AWS IoT polícticas de IAM. En este tema se analizan únicamente las políticas de IAM. Para obtener más información, consulte [Políticas de AWS IoT Core \(p. 333\)](#). AWS Identity and Access Management define una acción de política para cada operación que AWS IoT defina, incluidas las API de plano de control y de plano de datos.

Referencia de políticas gestionadas de IAM

AWS IoT proporciona un conjunto de políticas administradas de IAM que puede utilizar tal y como están, o bien como punto de partida para crear políticas de IAM personalizadas. Estas políticas le permiten tener acceso a operaciones de configuración y de datos. Las operaciones de configuración le permiten crear objetos, certificados, políticas y reglas. Las operaciones de datos envían datos sobre protocolos MQTT y HTTP. En la tabla siguiente se describen estas plantillas.

Plantilla de política	Descripción
AWSIoTConfigAccess	Permite a la identidad asociada tener acceso a todas las operaciones de configuración de AWS IoT. Esta política puede afectar al procesamiento y al almacenamiento de datos. Change log
AWSIoTConfigReadOnlyAccess	Permite a la identidad asociada obtener acceso a operaciones de configuración de solo lectura. Change log

Plantilla de política	Descripción
AWSIoTDataAccess	Permite a la identidad asociada tener acceso completo a todas las operaciones de datos de AWS IoT. Las operaciones de datos envían datos sobre protocolos MQTT y HTTP. Change log
AWSIoTEventsFullAccess	Permite el acceso completo de la identidad asociada a los eventos de AWS IoT. Change log
AWSIoTEventsReadOnlyAccess	Permite el acceso de solo lectura de la identidad asociada a los eventos de AWS IoT. Change log
AWSIoTFullAccess	Permite a la identidad asociada tener acceso completo a todas las operaciones de mensajería y de configuración de AWS IoT. Change log
AWSIoTLogging	Permite a la identidad asociada crear Amazon CloudWatch Registra grupos y transmite registros a los grupos. Esta política se asocia al CloudWatch rol de registro. Change log
AWSIoTOTUpdate	Permite crear la identidad asociada AWS IoT jobs, AWS IoT trabajos de firma de código y para describir AWS trabajos firmantes de código. Change log
AWSIoTRuleActions	Permite a la identidad asociada tener acceso a todas las AWS servicios compatibles en AWS IoT Acciones de reglas de.
AWSIoTThingsRegistration	Permite a la identidad asociada registrar objetos en bloque mediante la <code>StartThingRegistrationTask</code> API. Esta política puede afectar al procesamiento y al almacenamiento de datos. Change log
AWSIoTWirelessDataAccess	Permite a la identidad asociada enviar datos a AWS IoT Dispositivos inalámbricos. Change log

Plantilla de política	Descripción
AWSIoTWirelessFullAccess	Permite a la identidad asociada tener acceso completo a AWS IoTWireless. Change log
AWSIoTWirelessFullPublishAccess	Concesiones AWS IoT Acceso inalámbrico limitado para publicar en AWS IoT Reglas en su nombre. Change log
AWSIoTWirelessLogging	Permite a la identidad asociada crear Amazon CloudWatch grupos de registros y transmite registros a los grupos. Esta política se asocia al CloudWatch rol de registro. Change log
AWSIoTWirelessReadOnlyAccess	Permite a la identidad asociada tener acceso de solo lectura a AWS IoTWireless. Change log
AWSIoTWirelessGatewayCertManager	Permite a la identidad asociada acceso crear, enumerar y describir AWS IoT certificados de.
	Change log

Ejemplos de políticas basadas en identidad de AWS IoT

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

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

Temas

- [Prácticas recomendadas relativas a políticas \(p. 408\)](#)
- [Mediante la consola de AWS IoT \(p. 409\)](#)
- [Permitir a los usuarios consultar sus propios permisos \(p. 409\)](#)
- [Visualización de recursos de AWS IoT basados en etiquetas \(p. 410\)](#)
- [Visualización de AWS IoT Recursos de Device Advisor basados en etiquetas \(p. 411\)](#)

Prácticas recomendadas relativas a políticas

Las políticas basadas en identidades son muy eficaces. Determinan si alguien puede crear, acceder o eliminar los recursos de AWS IoT de su cuenta. Estas acciones pueden generar costes adicionales

para su Cuenta de AWS. Siga estas directrices y recomendaciones al crear o editar políticas basadas en identidades:

- Comenzar a utilizar políticas administradas de AWS: para comenzar a utilizar AWS IoT rápidamente, utilice las políticas administradas de AWS para proporcionar a los empleados los permisos necesarios. Estas políticas ya están disponibles en su cuenta, y AWS las mantiene y actualiza. Para obtener más información, consulte [Introducción sobre el uso de permisos con políticas administradas por AWS](#) en la Guía del usuario de IAM.
- Conceder privilegios mínimos: al crear políticas personalizadas, conceda solo los permisos necesarios para llevar a cabo una tarea. Comience con un conjunto mínimo de permisos y conceda permisos adicionales según sea necesario. Por lo general, es más seguro que comenzar con permisos que son demasiado tolerantes e intentar hacerlos más estrictos más adelante. Para obtener más información, consulte [Conceder privilegios mínimos](#) en la Guía del usuario de IAM.
- Habilitar la MFA para operaciones confidenciales: para mayor seguridad, obligue a los usuarios de IAM a utilizar la autenticación multifactor (MFA) para acceder a recursos u operaciones de API confidenciales. Para obtener más información, consulte [Uso de la autenticación multifactor \(MFA\) en AWS](#) en la Guía del usuario de IAM.
- Utilizar condiciones de política para mayor seguridad: en la medida en que sea práctico, defina las condiciones en las que las políticas basadas en identidad permitan el acceso a un recurso. Por ejemplo, puede escribir condiciones para especificar un rango de direcciones IP permitidas desde el que debe proceder una solicitud. También puede escribir condiciones para permitir solicitudes solo en un intervalo de hora o fecha especificado o para solicitar el uso de SSL o MFA. Para obtener más información, consulte [Elementos de la política de JSON de IAM: Condición](#)en la IAM User Guide.

Mediante la consola de AWS IoT

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

Para asegurarse de que esas entidades puedan seguir usando la consola de AWS IoT, asocie también la siguiente política administrada de AWS a las entidades: [AWSIoTFullAccess](#). Para obtener más información, consulte [Adición de permisos a un usuario](#)en la IAM User Guide.

No es necesario que conceda permisos mínimos para la consola a los usuarios que solo realizan llamadas a la AWS CLI o a la API de AWS. En su lugar, permite acceso únicamente a las acciones que coincidan con la operación de API que intenta realizar.

Permitir a los usuarios consultar sus propios permisos

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

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "ViewOwnUserInfo",  
            "Effect": "Allow",  
            "Action": [  
                "iam:GetUserPolicy",  
                "iam>ListGroupsForUser",  
                "iam>ListAttachedUserPolicies",  
                "iam:GetUser"
```

```
        "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": "*"
}
]
```

Visualización de recursos de AWS IoT basados en etiquetas

Puede utilizar las condiciones de su política basada en la identidad para controlar el acceso a los recursos de AWS IoT basados en etiquetas. En este ejemplo, se muestra cómo crear una política que permita visualizar un objeto. Sin embargo, los permisos solo se conceden si la etiqueta del objeto `Owner` tiene el valor del nombre de usuario de dicho usuario. Esta política también proporciona los permisos necesarios para llevar a cabo esta acción en la consola.

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

También puede asociar esta política al usuario de IAM en su cuenta. Si un usuario llamado `richard-roe` intenta ver un grupo de facturación de AWS IoT, el grupo de facturación debe estar etiquetado como `Owner=richard-roe` o como `owner=richard-roe`. De lo contrario, se le deniega el acceso. La clave de la etiqueta de condición `Owner` coincide con los nombres de las claves de condición `Owner` y `owner` porque no distinguen entre mayúsculas y minúsculas. Para obtener más información, consulte [Elementos de la política de JSON de IAM: Condición](#) en la [IAM User Guide](#).

Visualización de AWS IoT Recursos de Device Advisor basados en etiquetas

Puede utilizar las condiciones de su política basada en la identidad para controlar el acceso a AWS IoT Recursos de Device Advisor basados en etiquetas. El siguiente ejemplo muestra cómo crear una política que permita visualizar una definición de suite concreta. Sin embargo, los permisos solo se conceden si la etiqueta de definición de suite tiene `SuiteType` establecido en el valor de `MQTT`. Esta política también proporciona los permisos necesarios para llevar a cabo esta acción en la consola.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "ViewSuiteDefinition",  
            "Effect": "Allow",  
            "Action": "iotdeviceadvisor:GetSuiteDefinition",  
            "Resource": "arn:aws:iotdeviceadvisor:*:*:suitedefinition/*",  
            "Condition": {  
                "StringEquals": {"aws:ResourceTag/SuiteType": "MQTT"}  
            }  
        }  
    ]  
}
```

Solución de problemas de identidades y accesos en AWS IoT

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

Temas

- [No tengo autorización para realizar una acción en AWS IoT \(p. 411\)](#)
- [No tengo autorización para realizar la operación iam:PassRole \(p. 412\)](#)
- [Quiero ver mis claves de acceso \(p. 412\)](#)
- [Soy administrador y deseo permitir que otros obtengan acceso a AWS IoT \(p. 413\)](#)
- [Quiero permitir a personas externas a mi Cuenta de AWS el acceso a mis recursos de AWS IoT \(p. 413\)](#)

No tengo autorización para realizar una acción en AWS IoT

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

El siguiente error de ejemplo se produce cuando el usuario de IAM intenta utilizar la consola para ver detalles sobre una cosa, pero no tiene `iot:DescribeThing` permisos.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:  
    iot:DescribeThing  
        on resource: MyIoTThing
```

En este caso, Mateo pide a su administrador que actualice sus políticas de forma que pueda obtener acceso al recurso `MyIoTThing` mediante la acción `iot:DescribeThing`.

Uso de AWS IoT Device Advisor

Si utiliza AWS IoT Device Advisor, se produce el siguiente error de ejemplo cuando mateo jackson El usuario de IAM, intenta utilizar la consola para ver detalles sobre una definición de suite, pero no tiene iotdeviceadvisor: GetSuiteDefinition permisos.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:  
iotdeviceadvisor:GetSuiteDefinition  
on resource: MySuiteDefinition
```

En este caso, Mateo pide a su administrador que actualice sus políticas de forma que pueda obtener acceso a **MySuiteDefinition** recurso con el iotdeviceadvisor: **Definición de Get Suite** action.

No tengo autorización para realizar la operación iam:PassRole

Si recibe un error que indica que no está autorizado para llevar a cabo la acción iam:PassRole, debe ponerse en contacto con su administrador para recibir ayuda. Su administrador es la persona que le facilitó su nombre de usuario y contraseña. Pida a la persona que actualice sus políticas de forma que pueda transferir un rol a AWS IoT.

Algunos Servicios de AWS permiten transferir un rol existente al servicio en cuestión en lugar de crear un nuevo rol de servicio o rol vinculado a servicio. Para ello, debe tener permisos para transferir el rol al servicio.

En el siguiente ejemplo, el error se produce cuando un usuario de IAM denominado marymajor intenta utilizar la consola para realizar una acción en AWS IoT. Sin embargo, la acción requiere que el servicio cuente con permisos otorgados por un rol de servicio. Mary no tiene permisos para transferir el rol al servicio.

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

En este caso, Mary pide a su administrador que actualice sus políticas para que pueda realizar la acción iam:PassRole.

Quiero ver mis claves de acceso

Después de crear sus claves de acceso de usuario de IAM, puede ver su ID de clave de acceso en cualquier momento. Sin embargo, no puede volver a ver su clave de acceso secreta. Si pierde la clave de acceso secreta, debe crear un nuevo par de claves de acceso.

Las claves de acceso se componen de dos partes: un ID de clave de acceso (por ejemplo, AKIAIOSFODNN7EXAMPLE) y una clave de acceso secreta (por ejemplo, wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY). El ID de clave de acceso y la clave de acceso secreta se utilizan juntos, como un nombre de usuario y contraseña, para autenticar sus solicitudes. Administre sus claves de acceso con el mismo nivel de seguridad que para el nombre de usuario y la contraseña.

Important

No proporcione las claves de acceso a terceros, ni siquiera para que le ayuden a **buscar el ID de usuario canónico**. Si lo hace, podría conceder a otra persona acceso permanente a su cuenta.

Cuando cree un par de claves de acceso, se le pide que guarde el ID de clave de acceso y la clave de acceso secreta en un lugar seguro. La clave de acceso secreta solo está disponible en el momento de su creación. Si pierde la clave de acceso secreta, debe agregar nuevas claves de acceso a su usuario de IAM. Puede tener un máximo de dos claves de acceso. Si ya cuenta con dos, debe eliminar un par de

claves antes de crear uno nuevo. Para consultar las instrucciones, consulte [Administración de claves de acceso](#) en la Guía del usuario de IAM.

Soy administrador y deseo permitir que otros obtengan acceso a AWS IoT

Para permitir que otros obtengan acceso a AWS IoT, debe crear una entidad de IAM (usuario o rol) para la persona o la aplicación que necesita acceso. Esta persona utilizará las credenciales de la entidad para acceder a AWS. A continuación, debe asociar una política a la entidad que le conceda los permisos correctos en AWS IoT.

Para comenzar de inmediato, consulte [Creación del primer grupo y usuario delegado de IAM](#) en la Guía del usuario de IAM.

Quiero permitir a personas externas a mi Cuenta de AWS el acceso a mis recursos de AWS IoT

Puede crear un rol que los usuarios de otras cuentas o las personas externas a la organización puedan utilizar para acceder a sus recursos. Puede especificar una persona de confianza para que asuma el rol. En el caso de los servicios que admitan las políticas basadas en recursos o las listas de control de acceso (ACL), puede utilizar dichas políticas para conceder a las personas acceso a sus recursos.

Para obtener más información, consulte lo siguiente:

- Para obtener información acerca de si AWS IoT admite estas características, consulte [Cómo AWS IoT funciona con IAM \(p. 389\)](#).
- Para obtener información acerca de cómo proporcionar acceso a los recursos de las Cuentas de AWS de su propiedad, consulte [Proporcionar acceso a un usuario de IAM a otra Cuenta de AWS de la que es propietario](#) en la Guía del usuario de IAM.
- Para obtener información acerca de cómo proporcionar acceso a los recursos a Cuentas de AWS de terceros, consulte [Proporcionar acceso a Cuentas de AWS que son propiedad de terceros](#) en la Guía del usuario de IAM.
- Para obtener información sobre cómo proporcionar acceso mediante una identidad federada, consulte [Proporcionar acceso a usuarios autenticados externamente \(identidad federada\)](#) en la Guía del usuario de IAM.
- Para obtener información sobre la diferencia entre los roles y las políticas basadas en recursos para el acceso entre cuentas, consulte [Cómo los roles de IAM difieren de las políticas basadas en recursos](#) en la Guía del usuario de IAM.

Registro y monitorización

La monitorización es una parte importante del mantenimiento de la fiabilidad, la disponibilidad y el rendimiento de AWS IoT y sus soluciones de AWS. Debe recopilar datos de monitoreo de todas las partes de su solución de AWS para que le resulte más sencillo depurar cualquier error que se produzca en distintas partes del código, en caso de que ocurra. Para obtener información sobre los procedimientos de registro y monitoreo, consulte [Monitorización de AWS IoT \(p. 426\)](#)

Herramientas de monitorización

AWS proporciona herramientas que puede utilizar para monitorear AWS IoT. Puede configurar algunas de estas herramientas para que realicen la monitorización por usted. Algunas de las herramientas requieren intervención manual. Le recomendamos que automatice las tareas de monitorización en la medida de lo posible.

Herramientas de monitorización automatizadas

Puede utilizar las siguientes herramientas de monitorización automatizado para vigilar AWS IoT e informar cuando haya algún problema:

- Amazon CloudWatch Alarmas: observe una sola métrica durante el periodo que especifique especificado y realice una o varias acciones según el valor de la métrica relativo a un determinado umbral durante varios períodos de tiempo. La acción es una notificación enviada a un tema de Amazon Simple Notification Service (Amazon SNS) o a una política de Auto Scaling de Amazon EC2. CloudWatch Las alarmas de no invocan acciones simplemente porque se encuentren en determinado estado. El estado debe haber cambiado y debe mantenerse durante el número de periodos especificado. Para obtener más información, consulte [MonitorAWS IoTalarmas y métricas mediante Amazon CloudWatch \(p. 433\)](#).
- Amazon CloudWatch Registromonitoreo, almacene y tenga acceso a los archivos de registro de AWS CloudTrail otras fuentes. Amazon CloudWatch Los registros también le permiten ver los pasos críticosAWS IoTLos casos de prueba de Device Advisor toman, generan eventos y mensajes MQTT enviados desde sus dispositivos oAWS IoT Coredurante la ejecución de las pruebas. Estos registros permiten depurar y realizar acciones correctivas en sus dispositivos. Para obtener más información, consulte [MonitorAWS IoTcon CloudWatch Registros \(p. 449\)](#)Para obtener más información acerca de cómo utilizar Amazon CloudWatch, consulte [Supervisión de archivos de registroen laAmazon CloudWatch Guía del usuario de.](#)
- Amazon CloudWatch Eventos: seleccione los eventos y diríjalo a uno o varios flujos o funciones de destino para realizar cambios, capturar información de estado y aplicar medidas correctivas. Para obtener más información, consulte [Qué es Amazon CloudWatch Eventosen laAmazon CloudWatch Guía del usuario de.](#)
- AWS CloudTrailMonitoreo de registros de— Compartir archivos de registro entre cuentas, supervisar CloudTrail archivos de registro en tiempo real enviándolos a CloudWatch Registra, escribe aplicaciones de procesamiento de registros en Java y comprueba que sus archivos de registro no hayan cambiado después de que CloudTrail los entregue. Para obtener más información, consulte [Registre las llamadas a la API de AWS IoT con AWS CloudTrail. \(p. 469\)](#)y también [Uso de CloudTrail Archivos de logen laAWS CloudTrailGuía del usuario de.](#)

Herramientas de monitorización manual

Otra parte importante del monitoreoAWS IoTimplica la supervisión manual de los elementos que el CloudWatch las alarmas no cubren. LaAWS IoT, CloudWatch y otrosAWSlos paneles de consola de servicio proporcionan un at-a-glance vista del estado de suAWSentorno de. Le recomendamos que también compruebe los archivos de registro en AWS IoT.

- El panel de AWS IoT muestra lo siguiente:
 - Certificados de CA
 - Certificados
 - Políticas
 - Reglas
 - Objetos
- La página de inicio de CloudWatch muestra:
 - Alarmas y estado actual.
 - Gráficos de alarmas y recursos.
 - Estado de los servicios.

Puede usar CloudWatch para hacer lo siguiente:

- Crear [paneles personalizados](#) para monitorear los servicios que le interesan.
- Realizar un gráfico con los datos de las métricas para resolver problemas y descubrir tendencias.

- Buscar y examinar todas sus métricas de recursos de AWS.
- Crear y editar las alarmas de notificación de problemas.

Validación de la conformidad enAWS IoT Núcleo

Auditores externos evalúan la seguridad y la conformidad de los Servicios de AWS como parte de varios programas de conformidad de AWS, como SOC, PCI, FedRAMP e HIPAA.

Para saber si AWS IoT u otros Servicios de AWS están incluidos en el ámbito de programas de conformidad específicos, consulte [Servicios de AWS en el ámbito del programa de conformidad](#). Para obtener información general, consulte [Programas de conformidad de AWS](#).

Puede descargar los informes de auditoría de terceros utilizando AWS Artifact. Para obtener más información, consulte [Descarga de informes en AWS Artifact](#).

Su responsabilidad de conformidad al utilizar Servicios de AWS se determina en función de la sensibilidad de los datos, los objetivos de cumplimiento de su empresa y la legislación y los reglamentos correspondientes. AWS proporciona los siguientes recursos para ayudar con la conformidad:

- [Guías de inicio rápido de seguridad y conformidad](#): en estas guías de implementación, se analizan consideraciones de arquitectura y se proporcionan los pasos para implementar entornos de base de referencia en AWS que se centren en la seguridad y la conformidad.
- [Documento técnico sobre arquitectura para seguridad y conformidad de HIPAA](#): en este documento técnico, se describe cómo las empresas pueden utilizar AWS para crear aplicaciones aptas para HIPAA.

Note

No todos los Servicios de AWS son aptos para HIPAA. Para obtener más información, consulte la [Referencia de servicios aptos para HIPAA](#).

- [Recursos de conformidad de AWS](#): este conjunto de manuales y guías podría aplicarse a su sector y ubicación.
- [Evaluación de recursos con reglas](#) en la Guía para desarrolladores de AWS Config: el servicio AWS Config evalúa en qué medida las configuraciones de sus recursos cumplen las prácticas internas, las directrices del sector y las normativas.
- [AWS Security Hub](#): este Servicio de AWS proporciona una vista integral de su estado de seguridad en AWS que lo ayuda a verificar la conformidad con los estándares y las prácticas recomendadas del sector de seguridad.
- [AWS Audit Manager](#): este servicio de Servicio de AWS lo ayuda a auditar continuamente el uso de AWS con el fin de simplificar la forma en que administra el riesgo y la conformidad con las normativas y los estándares del sector.

Resiliencia enAWS Núcleo de IoT

La infraestructura global de AWS se divide en Región de AWS y zonas de disponibilidad. Las Región de AWS proporcionan varias zonas de disponibilidad físicamente independientes y aisladas que se encuentran conectadas mediante redes con un alto nivel de rendimiento y redundancia, además de baja latencia. Con las zonas de disponibilidad, puede diseñar y utilizar aplicaciones y bases de datos que realizan una comutación por error automática entre zonas de disponibilidad sin interrupciones. Las zonas de disponibilidad tienen una mayor disponibilidad, tolerancia a errores y escalabilidad que las infraestructuras tradicionales de centros de datos únicos o múltiples.

Para obtener más información acerca deRegión de AWSs y zonas de disponibilidad, consulte[AWS Infraestructura global](#).

AWS IoT Core guarda la información sobre los dispositivos en el registro de dispositivos. También almacena certificados de CA, certificados de dispositivo y datos de sombra de dispositivos. En caso de fallos de hardware o red, estos datos se replican automáticamente en las zonas de disponibilidad, pero no en todas las regiones.

AWS IoT Core publica eventos MQTT cuando se actualiza el registro de dispositivos. Puede utilizar estos mensajes para realizar una copia de seguridad de los datos del registro y guardarlos en algún lugar, como una tabla de DynamoDB. Usted es responsable de guardar los certificados que AWS IoT Core cree para usted o los que cree usted mismo. La sombra del dispositivo almacena datos de estado sobre sus dispositivos y se pueden volver a enviar cuando un dispositivo vuelve a conectarse. AWS IoT Device Advisor almacena información sobre la configuración del conjunto de pruebas. Estos datos se replican automáticamente en caso de fallos de hardware o red.

AWS IoT Core los recursos de son específicos de la región y no se replican entre regiones de AWSa menos que lo hagas específicamente.

Para obtener información acerca de las prácticas recomendadas de seguridad, consulte [Prácticas de seguridad recomendadas para AWS IoT Core \(p. 419\)](#).

Uso de AWS IoT Core con los puntos de enlace de la VPC de la interfaz

con AWS IoT Core, puedes crear [Puntos de conexión de datos de IoT](#) dentro de la VPC mediante [puntos de enlace de la VPC de interfaz](#). Los puntos de enlace de la VPC de tipo interfaz cuentan con AWS PrivateLink, una tecnología que puede utilizar para acceder a los servicios que se ejecutan en AWS mediante el uso de direcciones IP privadas. Para obtener más información, consulte [Amazon Virtual Private Cloud](#).

Para conectar dispositivos sobre el terreno en redes remotas, como una red corporativa a su AWS VPC, consulte las distintas opciones enumeradas en el [Matriz de conectividad de red a Amazon VPC](#).

Note

Los endpoints de VPC para IoT Core no se admiten actualmente en AWS Regiones de China.

Temas del capítulo:

- [Creación de puntos de enlace de la VPC para AWS IoT Core \(p. 416\)](#)
- [Control del acceso a AWS IoT Core sobre puntos de conexión de la VPC \(p. 417\)](#)
- [Limitaciones de los extremos de la VPC \(p. 418\)](#)
- [Escalado de puntos finales de VPC con IoT Core \(p. 418\)](#)
- [Uso de dominios personalizados con puntos de enlace de la VPC \(p. 418\)](#)
- [Disponibilidad de los puntos de enlace de la VPC para AWS IoT Core \(p. 418\)](#)

Creación de puntos de enlace de la VPC para AWS IoT Core

Para empezar a utilizar los endpoints de la VPC, simplemente [crear un punto de enlace de la VPC de tipo interfaz](#) y seleccione AWS IoT Core como el AWS service. Si está utilizando la CLI, primero llame `describe-vpc-endpoint-services` para asegurarse de que está eligiendo una zona de disponibilidad donde AWS IoT Core está presente en su particular Región de AWS. Por ejemplo, en us-east-1, este comando tendrá el siguiente aspecto:

```
aws ec2 describe-vpc-endpoint-services --service-name com.amazonaws.us-east-1.iot.data
```

Note

La función de VPC para crear automáticamente un registro DNS está deshabilitada porque el control de IoT y los extremos de datos están divididos. Para unirse a estos endpoints, yDebe crear manualmente un registro DNS privado. Para obtener más información acerca de los registros de DNS de VPC privados, consulte[DNS privado para puntos de enlace de interfaz](#). Para obtener más información acerca deAWS IoT CoreLimitaciones de la VPC, consulte[Limitaciones de los extremos de la VPC \(p. 418\)](#).

Para enrutar correctamente las consultas DNS desde sus dispositivos a las interfaces de endpoint de la VPC, debe crear manualmente registros DNS en una zona alojada privada que esté asociada a la VPC. Para empezar, consulte[Creación de una zona hospedada privada](#). Dentro de la zona alojada privada, cree un registro de alias para cada IP de elastic network interface para el extremo de la VPC. Si tiene varias IP de interfaz de red para varios endpoints de VPC, cree registros DNS ponderados con igual ponderación en todos los registros ponderados. Estas direcciones IP están disponibles en el[Descripción de las interfaces de red](#)Llamada a la API cuando se filtra por el ID del endpoint de la VPC en el campo de descripción.

Control del acceso aAWS IoT Coresobre puntos de conexión de la VPC

Puede restringir el acceso del dispositivo aAWS IoT Corepara permitirse únicamente a través del endpoint de la VPC mediante VPC[claves de contexto de condición](#).AWS IoT Coreadmite las siguientes claves de contexto relacionadas con la VPC:

- [SourceVpc](#)
- [SourceVpce](#)
- [VpcSourceIp](#)

Note

AWS IoT Coreno admite<https://docs.aws.amazon.com/vpc/latest/privatelink/vpc-endpoints-access.html#vpc-endpoint-policies>Políticas de endpoints de VPC en este momento.

Por ejemplo, la siguiente política concede permiso para conectarse aAWS IoT Coreutilizando un ID de cliente que coincide con el nombre de objeto y para publicar en cualquier tema que tenga como prefijo el nombre de objeto, condicionado a que el dispositivo se conecte a un punto de enlace de VPC con un ID de punto de enlace de VPC determinado. Esta política denegaría los intentos de conexión a su punto final de datos de IoT públicos.

```
{
    "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}/*"
    ]
}
}
```

Limitaciones de los extremos de la VPC

En esta sección se describen las limitaciones de los puntos de enlace de la VPC en comparación con los puntos de enlace públicos.

- Los puntos de enlace de la VPC se admiten actualmente para [Puntos de conexión de datos de IoT](#) sólo
- Los períodos de mantenimiento de MQTT están limitados a 230 segundos. Mantener vidas durante más tiempo que ese período se reducirá automáticamente a 230 segundos
- Cada extremo de VPC admite 100.000 dispositivos conectados simultáneamente en total. Si necesita más conexiones, consulte [Escalado de puntos finales de VPC con IoT Core \(p. 418\)](#).
- Los puntos de enlace de la VPC solo admiten el tráfico IPv4.
- Los puntos de conexión de la VPC servirán [Certificados ATS](#) sólo, excepto los dominios personalizados.
- [Políticas de punto de enlace de la VPC](#) no se admiten en este momento.

Escalado de puntos finales de VPC con IoT Core

AWS IoT Core Los endpoints de la VPC de interfaz están limitados a 100.000 dispositivos conectados en un único extremo de interfaz. Si su caso de uso requiere más conexiones simultáneas con el agente, le recomendamos utilizar varios endpoints de VPC y enrutar manualmente los dispositivos a través de los endpoints de la interfaz. Al crear registros DNS privados para enrutar el tráfico a los endpoints de la VPC, asegúrese de crear tantos registros ponderados como terminales de VPC para distribuir el tráfico entre los múltiples endpoints.

Uso de dominios personalizados con puntos de enlace de la VPC

Si desea utilizar dominios personalizados con puntos finales de VPC, debe crear los registros de nombres de dominio personalizados en una zona alojada privada y crear registros de enruteamiento en Route53. Para obtener más información, consulte [Creación de una zona hospedada privada](#).

Disponibilidad de los puntos de enlace de la VPC para AWS IoT Core

AWS IoT Core Los endpoints de VPC de interfaz están disponibles en todos los [AWS IoT Core](#) regiones admitidas, con la excepción de [AWS Regiones de China](#).

Seguridad de la infraestructura en AWS IoT

Como colección de servicios gestionados, AWS IoT está protegido por los procedimientos de seguridad de red globales que se describen en la [Amazon Web Services: Información general de los procesos de seguridad](#) Documento técnico.

Puede utilizar llamadas a la API publicadas en AWS para obtener acceso a AWS IoT a través de la red. Los clientes deben ser compatibles con Transport Layer Security (TLS) 1.2 o una versión posterior. Los clientes también deben ser compatibles con conjuntos de cifrado con confidencialidad directa total (PFS) tales como Ephemeral Diffie-Hellman (DHE) o Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). La mayoría de los sistemas modernos, como Java 7 y posteriores, son compatibles con estos modos. Para obtener más información, consulte [Seguridad de transporte en AWS IoT \(p. 382\)](#).

Las solicitudes deben estar firmadas mediante un ID de clave de acceso y una clave de acceso secreta que esté asociada a una entidad principal de IAM. También puede utilizar [AWS Security Token Service \(AWS STS\)](#) para generar credenciales de seguridad temporales para firmar solicitudes.

Supervisión de seguridad de flotas o dispositivos de producción con AWS IoT Núcleo

Las flotas de IoT pueden constar de un gran número de dispositivos que tienen diversas funcionalidades, son de larga duración y están distribuidos geográficamente. Estas características hacen compleja la configuración de la flota y propenso a errores. Y dado que los dispositivos a menudo están limitados en potencia informática, memoria y capacidades de almacenamiento, esto limita el uso del cifrado y otras formas de seguridad en los propios dispositivos. Además, los dispositivos a menudo usan software con vulnerabilidades conocidas. Estos factores hacen que las flotas de IoT sean un objetivo atractivo para los piratas informáticos y dificultan la protección continuada de la flota de dispositivos.

Para afrontar estos desafíos, AWS IoT Device Defender proporciona herramientas para identificar los problemas de seguridad y las desviaciones de las prácticas recomendadas. Puede usar AWS IoT Device Defender para analizar, auditar y supervisar los dispositivos conectados para detectar comportamientos anómalos y mitigar los riesgos de seguridad. AWS IoT Device Defender puede auditar las flotas de dispositivos para asegurarse de que cumplan con las prácticas recomendadas de seguridad y detecten comportamientos anómalos en los dispositivos. Esto posibilita aplicar políticas de seguridad coherentes en toda su flota de dispositivos AWS IoT y responder rápidamente cuando los dispositivos sufren ataques. Para obtener más información, consulte [AWS IoT Device Defender \(p. 871\)](#).

AWS IoT Device Advisor envía actualizaciones y revisa su flota según sea necesario. AWS IoT Device Advisor actualiza los casos de prueba automáticamente. Los casos de prueba están siempre en la última versión. Para obtener más información, consulte [Asesor de dispositivos \(p. 1058\)](#).

Prácticas de seguridad recomendadas para AWS IoT Core

Esta sección contiene información sobre las prácticas de seguridad recomendadas para AWS IoT Core. Para obtener más información, consulte [Ten security golden rules for IoT solutions](#).

Protección de conexiones MQTT en AWS IoT

AWS IoT Core es un servicio administrado en la nube que permite que los dispositivos conectados interactúen de forma sencilla y segura con las aplicaciones en la nube. AWS IoT Core admite HTTP,

[WebSocket](#) y [MQTT](#), un protocolo de comunicación ligero especialmente diseñado para admitir conexiones intermitentes. Si te estás conectando aAWS IoTCon MQTT, cada una de sus conexiones debe asociarse con un identificador conocido como ID de cliente. Los ID de cliente MQTT identifican de forma exclusiva conexiones MQTT. Si se establece una nueva conexión mediante un ID de cliente que ya se ha solicitado para otra conexión, el agente de mensajes de AWS IoT interrumpe la conexión anterior para permitir la nueva conexión. Los ID de cliente deben ser únicos dentro de cadaCuenta de AWSy cadaRegión de AWS. Esto significa que no es necesario imponer la exclusividad global de los identificadores de cliente fuera de suCuenta de AWSen distintas regiones dentro de tuCuenta de AWS.

El impacto y la gravedad de la interrupción de las conexiones MQTT en su flota de dispositivos depende de muchos factores. Entre ellas se incluyen:

- Su caso de uso (por ejemplo, los datos que los dispositivos envían a AWS IoT, la cantidad de datos y la frecuencia con la que se envían).
- Su configuración de cliente MQTT (por ejemplo, configuración de reconexión automática, temporizaciones de interrupción asociadas y uso de [sesiones persistentes de MQTT \(p. 86\)](#)).
- Las restricciones de recursos de dispositivos.
- La causa principal de las desconexiones, su agresividad y persistencia.

Para evitar conflictos de ID de cliente y sus posibles efectos negativos, asegúrese de que cada dispositivo o aplicación móvil tenga unaAWS IoTpolítica de IAM que restringe qué ID de cliente se pueden utilizar para las conexiones MQTT aAWS IoTagente de mensajes. Por ejemplo, puede utilizar una política de IAM para evitar que un dispositivo cierre por error la conexión de otro dispositivo utilizando un ID de cliente que ya está en uso. Para obtener más información, consulte [Autorización \(p. 331\)](#).

Todos los dispositivos de la flota deben tener credenciales con privilegios que autoricen únicamente las acciones previstas; por ejemplo, acciones MQTT de AWS IoT, como la publicación de mensajes o la suscripción a temas con un ámbito y un contexto específicos. Las políticas de permisos específicas pueden variar en función de los casos de uso. Identifique las políticas de permisos que se ajusten mejor a sus requisitos empresariales y de seguridad.

Para simplificar la creación y la administración de políticas de permisos, puede utilizar[Variables de las políticas de AWS IoT Core \(p. 337\)](#)y[Variables de política de IAM](#). Las variables de la política se pueden colocar en una política y cuando se evalúa la política las variables se sustituyen por valores procedentes de la solicitud del dispositivo. Al usar variables de la política, puede crear una única política para la concesión de permisos a varios dispositivos. Puede identificar las variables de la política relevantes para su caso de uso en función de su configuración de la cuenta de AWS IoT, el mecanismo de autenticación y el protocolo de red utilizado en la conexión al agente de mensajes de AWS IoT. Sin embargo, para escribir las mejores políticas de permisos, deben tenerse en cuenta los detalles concretos de su caso de uso y el [modelo de amenazas](#).

Por ejemplo, si registraste tus dispositivos en elAWS IoTregistro, puedes usar[Variables de política de objeto \(p. 339\)](#)enAWS IoTpolíticas de concesión o denegación de permisos en función de las propiedades del objeto, como el nombre o el tipo de objeto, o los valores de atributo del objeto. El nombre del objeto se obtiene a partir del ID de cliente del mensaje de MQTT Connect que se envía cuando un objeto se conecta a AWS IoT. Las variables de política de cosa se reemplazan cuando una cosa se conecta aAWS IoTa través de MQTT mediante autenticación mutua TLS o MQTT a través del WebSocket protocolo mediante autenticación de Amazon Cognito. Puede utilizar el[AttachThingPrincipalAPI](#) para asociar certificados e identidades de Amazon Cognito autenticadas a una cosa.iot : Connection . Thing . ThingNamees una variable de política de objetos útil para forzar la aplicación de las restricciones del ID de cliente. La siguiente política de AWS IoT de ejemplo requiere un nombre del objeto registrado que se va a utilizar como el ID de cliente para las conexiones MQTT al agente de mensajes de 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}"  
    ]  
}  
}  
}
```

Si desea identificar conflictos de ID de cliente en curso, puede habilitar y utilizar[CloudWatch Logs paraAWS IoT \(p. 449\)](#). Por cada conexión MQTT que el agente de mensajes de AWS IoT desconecta debido a conflictos de ID de cliente, se genera un registro parecido a lo siguiente:

```
{  
    "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"  
}
```

Puede usar un[Filtro CloudWatch Logstales como{ \\$.reason= "DUPLICATE_CLIENT_ID" }](#) buscar instancias de conflictos de ID de cliente o configurar[Filtros métricos CloudWatchy](#) correspondiente CloudWatch alarmas para monitoreo e informes continuos.

Puede usar[AWS IoTDevice Defender](#)identificar excesivamente permisivoAWS IoT polticas de IAM.AWS IoT Device Defender tambin proporciona una comprobacin de auditora que le notifica si varios dispositivos en la flota se conecta alAWS IoTagente de mensajes con el mismo ID de cliente.

Puede usar[AWS IoTDevice Advisor](#) para validar que sus dispositivos se pueden conectar de forma fiable aAWS IoT Corey siga las prcticas recomendadas de seguridad de.

Véase tambén

- [AWS IoT Core](#)
- [Características de seguridad de AWS IoT \(p. 294\)](#)
- [Variables de las polticas de AWS IoT Core \(p. 337\)](#)
- [Variables de poltica de IAM](#)
- [Identidad de Amazon Cognito](#)
- [AWS IoT Device Defender](#)
- [CloudWatch Logs paraAWS IoT \(p. 449\)](#)

Mantener sincronizado el reloj del dispositivo

Es importante que la hora del dispositivo sea precisa. Los certificados X.509 tienen una fecha y una hora de caducidad. El reloj del dispositivo se utiliza para comprobar que un certificado de servidor sigue siendo

válido. Si está creando dispositivos IoT comerciales, recuerde que sus productos pueden permanecer en el almacén durante largos períodos de tiempo antes de venderse. Los relojes en tiempo real pueden retrasarse durante este tiempo y las baterías pueden descargarse, por lo que no es suficiente fijar el tiempo en fábrica.

En la mayoría de los sistemas, esto significa que el software del dispositivo debe incluir un cliente NTP (Protocolo de tiempo de red). El dispositivo debe esperar hasta que se sincronice con un servidor NTP antes de intentar conectarse a AWS IoT Core. Si esto no es posible, el sistema debería proporcionar un mecanismo para que el usuario establezca la hora del dispositivo, de forma que las conexiones posteriores se realicen correctamente.

Una vez que el dispositivo se haya sincronizado con un servidor NTP, podrá abrir una conexión con AWS IoT Core. La cantidad de sesgo de reloj permitida dependerá de lo que esté tratando de hacer con la conexión.

Validar el certificado de servidor

Lo primero que un dispositivo hace para interactuar con AWS IoT es abrir una conexión segura. Cuando conecte el dispositivo a AWS IoT, asegúrese de que está hablando con otro servidor AWS IoT y no a otro servidor que esté suplantando la identidad de AWS IoT. Cada uno de los servidores de AWS IoT se aprovisiona con un certificado emitido para el dominio `iot.amazonaws.com`. Este certificado fue emitido para AWS IoT por una autoridad de certificación de confianza que verificó nuestra identidad y propiedad del dominio.

Una de las primeras cosas que AWS IoT Core hace cuando se conecta un dispositivo es enviar al dispositivo un certificado de servidor. Los dispositivos pueden comprobar que estaban esperando para conectarse a `iot.amazonaws.com` y que el servidor al final de dicha conexión posee un certificado de una autoridad de confianza para ese dominio.

Los certificados TLS tienen formato X.509 e incluyen información diversa, como el nombre de la organización, la ubicación, el nombre de dominio y un período de validez. El período de validez se especifica como un par de valores de tiempo llamados `notBefore` y `notAfter`. Algunos servicios como AWS IoT Core utilizan períodos de validez limitados (por ejemplo, un año) para los certificados de servidor y comienzan a proporcionar otros nuevos antes de que caduquen los antiguos.

Usar una identidad única por dispositivo

Utilice una identidad única por cliente. Los dispositivos generalmente usan certificados de cliente X.509. Las aplicaciones web y móviles utilizan Amazon Cognito Identity. Esto le permite aplicar permisos detallados a sus dispositivos.

Por ejemplo, tiene una aplicación que consiste en un dispositivo de teléfono móvil que recibe actualizaciones de estado de dos objetos de hogar inteligente diferentes: una bombilla y un termostato. La bombilla envía el estado de su nivel de batería y el termostato envía mensajes que informan de la temperatura.

AWS IoT autentica los dispositivos por separado y trata cada conexión individualmente. Puede aplicar controles de acceso detallados a través de políticas de autorización. Puede definir una política para el termostato que le permita publicar en un espacio de tema. Puede definir una política independiente para la bombilla que le permita publicar en un espacio de tema diferente. Por último, puede definir una política para la aplicación móvil que solo le permita conectarse y suscribirse a los temas del termostato y la bombilla para recibir mensajes de estos dispositivos.

Aplique el principio de privilegios mínimos y amplíe los permisos por dispositivo tanto como sea posible. Todos los dispositivos o usuarios deben tener una política de AWS IoT en AWS IoT que solo les permita conectarse con un ID de cliente conocido, así como publicar y suscribirse a un conjunto de temas identificado y fijo.

Utilice un segundoRegión de AWScomo backup

Considere almacenar una copia de sus datos en un segundoRegión de AWScomo backup. Para obtener más información, consulte [Recuperación de desastres paraAWS IoT](#).

Usar aprovisionamiento justo a tiempo

Crear y aprovisionar manualmente cada dispositivo puede llevar mucho tiempo. AWS IoT proporciona una forma de definir una plantilla para aprovisionar dispositivos cuando se conectan por primera vez a AWS IoT. Para obtener más información, consulte [Aprovisionamiento justo a tiempo \(p. 804\)](#).

Permisos para ejecutarAWS IoTPrueba de Device Advisor

La siguiente plantilla de política muestra los permisos mínimos y la entidad de IAM necesarios para ejecutarAWS IoTCasos de prueba de Device Advisor. Tendrás que sustituir *your-device-role-arn* con el nombre de recurso de Amazon (ARN) del rol de dispositivo que creó en la Requisitos previos de.

```
{
    "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": [
                "iam>ListRoles", // Required to list device roles in the device advisor
                "iot:Connect",
                "logs:DescribeLogStreams",
                "iot:DescribeThing",
                "iot:DescribeCertificate",
                "logs>CreateLogGroup",
                "logs:DescribeLogGroups",
                "logs:PutLogEvents",
                "iot:DescribeEndpoint",
                "execute-api:Invoke*",
                "logs>CreateLogStream",
                "iot>ListPrincipalPolicies",
                "iot>ListThingPrincipals",
                "iot>ListThings",
                "iot:Publish",
                "iot>CreateJob",
                "iot:DescribeJob",
                "iot>ListCertificates",
                "iot>ListAttachedPolicies",
                "iot:UpdateThingShadow",
                "iot:GetPolicy"
            ],
        }
    ]
}
```

```
        "Resource": "*"
    },
{
    "Sid": "VisualEditor2",
    "Effect": "Allow",
    "Action": "iotdeviceadvisor:/*",
    "Resource": "*"
}
]
```

Prevención del suplente confuso entre servicios para Device Advisor

El problema de la sustitución confusa es una cuestión de seguridad en la que una entidad que no tiene permiso para realizar una acción puede obligar a una entidad con más privilegios a realizar la acción. En AWS, la suplantación entre servicios puede dar lugar al problema del suplente confuso. La suplantación entre servicios puede producirse cuando un servicio (el servicio que lleva a cabo las llamadas) llama a otro servicio (el servicio al que se llama). El servicio que lleva a cabo las llamadas se puede manipular para utilizar sus permisos a fin de actuar en función de los recursos de otro cliente de una manera en la que no debe tener permiso para acceder. Para evitarlo, AWS proporciona herramientas que lo ayudan a proteger sus datos para todos los servicios con entidades principales de servicio a las que se les ha dado acceso a los recursos de su cuenta.

Le recomendamos que utilice la `aws:SourceArn` y `aws:SourceAccount` claves de contexto de condición global en las políticas de recursos para limitar los permisos que Device Advisor concede a otro servicio para el recurso. Si se utilizan ambas claves de contexto de condición global, el valor `aws:SourceAccount` y la cuenta del valor `aws:SourceArn` deben utilizar el mismo ID de cuenta cuando se utilicen en la misma declaración de política.

El valor de `aws:SourceArn` debe ser el ARN de su recurso de definición de suite. El recurso de definición de suite hace referencia al conjunto de pruebas que creó con Device Advisor.

La forma más eficaz de protegerse contra el problema del suplente confuso es utilizar la clave de contexto de condición global de `aws:SourceArn` con el ARN completo del recurso. Si no conoce el ARN completo del recurso o si está especificando varios recursos, utilice la clave de condición de contexto global `aws:SourceArn` con comodines (*) para las partes desconocidas del ARN. Por ejemplo, `arn:aws:iotdeviceadvisor:*:account-id:suitedefinition/*`.

El siguiente ejemplo muestra cómo puede utilizar la `aws:SourceArn` y `aws:SourceAccount` claves de contexto de condición global en Device Advisor para evitar el problema confundido del adjunto.

```
{
    "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/ygp6rxax3tzvn"
            },
            "StringEquals": {
                "aws:SourceAccount": "123456789012"
            }
        }
}
```



AWS Training and Certification

Siga el siguiente curso para conocer los conceptos clave de AWS IoT seguridad:[AWS IoT Manual de seguridad](#).

Monitorización de AWS IoT

La monitorización es una parte importante del mantenimiento de la fiabilidad, la disponibilidad y el rendimiento de AWS IoT y sus soluciones de AWS.

Le recomendamos encarecidamente que recopile datos de monitoreo de todas las partes de su solución de AWS para que le resulte más sencillo depurar cualquier error que se produzca en distintas partes del código, en caso de que ocurra. Comience por crear un plan de monitoreo que responda a las siguientes preguntas. Si no está seguro de cómo responderlas, puede continuar [habilitando el registro \(p. 427\)](#) y estableciendo las líneas base de rendimiento.

- ¿Cuáles son los objetivos de la monitorización?
- ¿Qué recursos va a monitorizar?
- ¿Con qué frecuencia va a monitorizar estos recursos?
- ¿Qué herramientas de monitorización va a utilizar?
- ¿Quién se encargará de realizar las tareas de monitoreo?
- ¿Quién debería recibir una notificación cuando surjan problemas?

El siguiente paso consiste en [habilitar el registro \(p. 427\)](#) y establecer un punto de referencia de rendimiento de AWS IoT normal en el entorno midiendo el rendimiento varias veces y bajo distintas condiciones de carga. A medida que monitorea AWS IoT, mantenga los datos de monitoreo históricos para que pueda compararlos con los datos de rendimiento actuales. Esto le ayudará a identificar patrones de rendimiento normales y anomalías de rendimiento, así como a idear métodos para abordarlos.

Para establecer el rendimiento previsto para AWS IoT, debe monitorear estas métricas para comenzar. Siempre puede monitorear más métricas más adelante.

- [PublishIn.Success \(p. 440\)](#)
- [PublishOut.Success \(p. 440\)](#)
- [Subscribe.Success \(p. 440\)](#)
- [Ping.Success \(p. 440\)](#)
- [Connect.Success \(p. 440\)](#)
- [GetThingShadow.Accepted \(p. 442\)](#)
- [UpdateThingShadow.Accepted \(p. 442\)](#)
- [DeleteThingShadow.Accepted \(p. 442\)](#)
- [RulesExecuted \(p. 438\)](#)

Los temas de esta sección pueden ayudarle a iniciar el registro y el monitoreo de AWS IoT.

Temas

- [Configuración de registros de AWS IoT \(p. 427\)](#)
- [MonitorAWS IoT alarmas y métricas mediante Amazon CloudWatch \(p. 433\)](#)
- [MonitorAWS IoT con CloudWatch Registros \(p. 449\)](#)
- [Registre las llamadas a la API de AWS IoT con AWS CloudTrail. \(p. 469\)](#)

Configuración de registros de AWS IoT

Debe habilitar el registro mediante la consola de AWS IoT, la CLI o la API antes de poder monitorear y registrar la actividad de AWS IoT.

Puede habilitar el registro para todo AWS IoT o solo para grupos de objetos específicos. Puede configurar el registro de AWS IoT mediante la consola de AWS IoT, la CLI o la API; sin embargo, debe usar la CLI o la API para configurar el registro para grupos de objetos específicos.

Al considerar cómo configurar el registro de AWS IoT, la configuración predeterminada de registro determina cómo se registrará la actividad de AWS IoT a menos que se especifique lo contrario.

Para empezar, es posible que desee obtener registros detallados con un [nivel de registro \(p. 432\)](#) predeterminado de `INFO` o `DEBUG`. Después de revisar los registros iniciales, puede cambiar el nivel de registro predeterminado a un nivel menos detallado, como `WARN` o `ERROR` y establecer un nivel de registro específico de recursos más detallado en los recursos que puedan necesitar más atención. Los niveles de registro se pueden cambiar cuando lo deseé.

Configuración del rol y la política de registro

Antes de habilitar el inicio de sesión AWS IoT, debe crear un rol de IAM y una política que dé AWS permiso para supervisar AWS IoT actividad en su nombre.

Note

Antes de activar AWS IoT Registro, asegúrese de comprender bien la CloudWatch Registro los permisos de acceso. Usuarios con acceso a CloudWatch Los registros de pueden consultar la información de depuración de sus dispositivos. Para obtener más información, consulte [Autenticación y control de acceso de Amazon CloudWatch Registros](#).

Si espera patrones de tráfico elevado en AWS IoT Core debido a las pruebas de carga, considere desactivar el registro de IoT para evitar la estrangulación. Si se detecta un tráfico elevado, nuestro servicio puede deshabilitar el inicio de sesión en su cuenta.

A continuación se muestra cómo crear un rol y una política de registro para AWS IoT Core de AWS. Para obtener información sobre cómo crear un rol y una política de registro de IAM para AWS IoT Core o LoRaWAN, consulte [Creación de una política y un rol de registro para AWS IoT Wireless \(p. 1221\)](#).

Creación de un rol de registro

Para crear un rol de registro, abra la [Centro de roles de la consola de IAM](#) y elige Creación de un rol.

1. UNDER Seleccione el tipo de entidad de confianza, elige AWS Service (Servicio), IoT.
2. UNDER Seleccione su caso de uso, eligelo Ty luego seleccione Siguiente: Permisos.
3. En la página que muestra las políticas que se asocian automáticamente al rol de servicio, elija Siguiente: Etiquetas y luego seleccione Siguiente: Consulte.
4. Escriba un Nombre de rol y una Descripción de rol para el rol y, a continuación, elija Crear rol.
5. En la lista de Roles de, busque el rol que creó, ábralo y copie el ARN de rol (`arn de role-arn`) para usar cuando Configure el registro predeterminado en AWS IoT (consola) (p. 428).

Política de rol de registro

Los documentos de política siguientes proporcionan la política de rol y de confianza que permiten AWS IoT para enviar entradas de registro a CloudWatch en su nombre. Si también te lo permite AWS IoT Core para que LoRaWAN envíe entradas de registro, verá un documento de política creado para usted que registra ambas actividades. Para obtener información acerca de cómo crear un rol y una política de registro de IAM para AWS IoT Core o LoRaWAN, consulte [Creación de una política y un rol de registro para AWS IoT Wireless \(p. 1221\)](#).

Note

Estos documentos se crearon para usted cuando creó el rol de registro.

Política de roles:

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "logs:CreateLogGroup",  
                "logs:CreateLogStream",  
                "logs:PutLogEvents",  
                "logs:PutMetricFilter",  
                "logs:PutRetentionPolicy"  
            ],  
            "Resource": [  
                "*"  
            ]  
        }  
    ]  
}
```

Política de confianza para iniciar sesión únicamenteAWS IoT Core actividad:

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

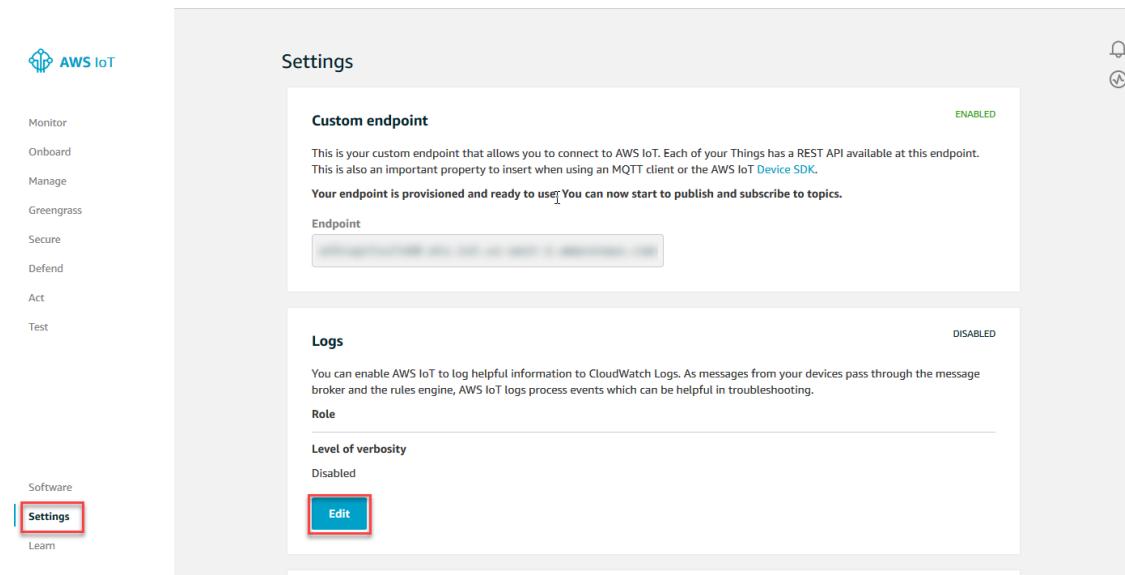
Configure el registro predeterminado en AWS IoT (consola)

En esta sección se describe cómo utilizar la consola de AWS IoT para configurar el registro para todo AWS IoT. Para configurar el registro solo para grupos de objetos específicos, debe usar la CLI o la API. Para obtener información sobre cómo configurar el registro para grupos de objetos específicos, consulte [Configurar el inicio de sesión específico de recursos en AWS IoT \(CLI\) \(p. 431\)](#).

Para usar la consola de AWS IoT para configurar el registro predeterminado para todo AWS IoT

1. Inicie sesión en la consola de AWS IoT. Para obtener más información, consulte [Abra el iconoAWS IoT consola \(p. 20\)](#).
2. En el panel de navegación izquierdo, elija Configuración. En la sección Registros de la página Configuración, elija Editar.

La sección Registros muestra el rol de registro y el nivel de detalle utilizado por todo AWS IoT.



3. En la página Configuración de la configuración de roles, seleccione la nivel de detalle que describe la [nivel de detalle \(p. 432\)](#) de las entradas de registro que desea que aparezcan en el CloudWatch registros.

The screenshot shows the 'Configure role setting' dialog box. It has a blue header bar with the title. Below it, there's a 'Level of verbosity' section with a dropdown menu set to 'Disable logging'. A note says: 'There are four levels of log verbosity. For example, you can choose "Errors" to get only logs about errors, or "Info" to get informational logs, warnings, and errors. [Click here](#) to learn more about troubleshooting in AWS IoT with CloudWatch logs.' In the 'Set role' section, it says 'You can select a role to log specific account-level information to CloudWatch Logs.' There are two buttons: 'Create Role' and 'Select'. At the bottom, there are 'Cancel' and 'Update' buttons, with 'Update' highlighted with a red box.

4. Elija Seleccionar para especificar un rol que creó en [Creación de un rol de registro \(p. 427\)](#) o Crear rol para crear un nuevo rol para utilizarlo para registro.
5. Elija Actualizar para guardar los cambios.

Después de habilitar el registro, visite [Visualización de AWS IoT inicio de sesión en el CloudWatch consola \(p. 449\)](#) para obtener más información sobre cómo ver las entradas del registro.

Configurar el registro predeterminado en AWS IoT (CLI)

En esta sección se describe cómo configurar el registro global para AWS IoT mediante la CLI.

Note

Necesita el nombre de recurso de Amazon (ARN) del rol que desea utilizar. Si necesita crear un rol para usar en el registro, consulte [Creación de un rol de registro \(p. 427\)](#) antes de continuar. La entidad principal que se utiliza para realizar la llamada a la API debe tener [Transmisión de los permisos de rol \(p. 474\)](#) para el rol de registro.

También puede realizar este procedimiento con la API utilizando los métodos de la API de AWS que corresponden a los comandos CLI que se muestran aquí.

Para usar la CLI para configurar el registro predeterminado para AWS IoT

1. Utilice el comando [set-v2-logging-options](#) para establecer las opciones de registro para su cuenta.

```
aws iot set-v2-logging-options \
    --role-arn logging-role-arn \
    --default-log-level log-level
```

donde:

--role-arn

El ARN del rol que concede AWS IoT permiso para escribir en los registros CloudWatch Registro.
--default-log-level

El [nivel de registro \(p. 432\)](#) que se debe usar. Los valores válidos son: ERROR, WARN, INFO, DEBUG o DISABLED

--no-disable-all-logs

Un parámetro opcional que habilita todo el registro de AWS IoT. Utilice este parámetro para habilitar el registro cuando esté deshabilitado actualmente.

--disable-all-logs

Un parámetro opcional que deshabilita todo el registro de AWS IoT. Utilice este parámetro para deshabilitar el registro cuando esté habilitado actualmente.

2. Utilice el comando [get-v2-logging-options](#) para obtener las opciones de registro actuales.

```
aws iot get-v2-logging-options
```

Después de habilitar el registro, visite [Visualización de AWS IoT inicio de sesión en el CloudWatch consola \(p. 449\)](#) para obtener más información sobre cómo ver las entradas del registro.

Note

AWS IoT sigue siendo compatible con los comandos más antiguos (set-logging-options y get-logging-options) para establecer y obtener el registro global en su cuenta. Tenga en cuenta que, cuando se utilizan estos comandos, los registros resultantes contienen texto sin formato, en lugar de cargas JSON y que la latencia de registro normalmente es mayor. No se realizarán mejoras en la implementación de estos comandos más antiguos. Le recomendamos que utilice las versiones "v2" para configurar las opciones de registro y, cuando sea posible, que modifique las aplicaciones heredadas que utilizan las versiones más antiguas.

Configurar el inicio de sesión específico de recursos en AWS IoT (CLI)

En esta sección se describe cómo configurar el registro específico de recursos para AWS IoT mediante la CLI. El registro específico de recursos le permite especificar un nivel de registro para un [grupo de objetos \(p. 273\)](#) específico.

Los grupos de objetos pueden contener otros grupos de objetos para crear una relación jerárquica. Este procedimiento describe cómo configurar el registro de un solo grupo de objetos. Puede aplicar este procedimiento al grupo de objetos principal de una jerarquía para configurar el registro de todos los grupos de objetos de la jerarquía. También puede aplicar este procedimiento a un grupo de objetos secundario para anular la configuración de registro de su principal.

Además de los grupos de cosas, también puede registrar destinos como el ID de cliente de un dispositivo, la IP de origen y el ID principal.

Note

Necesita el nombre de recurso de Amazon (ARN) del rol que desea utilizar. Si necesita crear un rol para usar en el registro, consulte [Creación de un rol de registro \(p. 427\)](#) antes de continuar.

La entidad principal que se utiliza para realizar la llamada a la API debe tener [Transmisión de los permisos de rol \(p. 474\)](#) para el rol de registro.

También puede realizar este procedimiento con la API utilizando los métodos de la API de AWS que corresponden a los comandos CLI que se muestran aquí.

Para usar la CLI para configurar el registro específico de recursos para AWS IoT

1. Utilice el comando [set-v2-logging-options](#) para establecer las opciones de registro para su cuenta.

```
aws iot set-v2-logging-options \
    --role-arn logging-role-arn \
    --default-log-level log-level
```

donde:

--role-arn

El ARN del rol que concede AWS IoT permiso para escribir en los registros CloudWatch Registros.

--default-log-level

El [nivel de registro \(p. 432\)](#) que se debe usar. Los valores válidos son: ERROR, WARN, INFO, DEBUG o DISABLED

--no-disable-all-logs

Un parámetro opcional que habilita todo el registro de AWS IoT. Utilice este parámetro para habilitar el registro cuando esté deshabilitado actualmente.

--disable-all-logs

Un parámetro opcional que deshabilita todo el registro de AWS IoT. Utilice este parámetro para deshabilitar el registro cuando esté habilitado actualmente.

2. Utilice el comando [set-v2-logging-level](#) para configurar el registro específico de recursos para un grupo de objetos.

```
aws iot set-v2-logging-level \
    --log-target targetType=THING_GROUP,targetName=thing_group_name \
```

```
--log-level log_level
```

--log-target

El tipo y nombre del recurso para el que está configurando el registro.

La target_type value debe ser una de las siguientes opciones: THING_GROUP | CLIENT_ID | SOURCE_IP | PRINCIPAL_ID. El valor del parámetro log-target puede ser texto, como se muestra en el ejemplo de comando anterior, o una cadena JSON, como en el ejemplo siguiente.

```
aws iot set-v2-logging-level \  
    --log-target '{"targetType": "THING_GROUP", "targetName": "thing_group_name"}' \  
    --log-level log_level
```

--log-level

El nivel de registro utilizado cuando se generan registros para el recurso especificado. Los valores válidos son: DEBUG, INFO, ERROR, WARN y DISABLED.

```
aws iot set-v2-logging-level \  
    --log-target targetType=CLIENT_ID,targetName=ClientId1 \  
    --log-level DEBUG
```

3. Utilice el comando [list-v2-logging-levels](#) para enumerar los niveles de registro configurados actualmente.

```
aws iot list-v2-logging-levels
```

4. Usar [delete-v2-logging-level](#) para eliminar un nivel de registro específico de recursos, como los ejemplos siguientes.

```
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

La target_type value debe ser una de las siguientes opciones: THING_GROUP | CLIENT_ID | SOURCE_IP | PRINCIPAL_ID.

--targetName

El nombre del grupo de objetos para el que se va a quitar el nivel de registro.

Después de habilitar el registro, visite [Visualización de AWS IoT en el CloudWatch consola \(p. 449\)](#) para obtener más información sobre cómo ver las entradas del registro.

Niveles de registro

Estos niveles de registro determinan los eventos que se registran y se aplican a los niveles de registro predeterminados y específicos de recursos.

ERROR

Cualquier error que provoque el fracaso de una operación.

Los registros solo incluirán información de ERROR.

WARN

Todo lo que pueda llegar a producir incoherencias en el sistema, aunque no el fracaso de la operación.

Los registros incluirán información de ERROR y WARN.

INFO

Información general acerca del flujo de objetos.

Los registros incluirán información de INFO, ERROR y WARN.

DEBUG

Información que puede ser útil para depurar un problema.

Los registros incluirán información de DEBUG, INFO, ERROR y WARN.

DISABLED

Todos los registros están desactivados.

MonitorAWS IoT alarmas y métricas mediante Amazon CloudWatch

Puede monitorear AWS IoT mediante CloudWatch, que recopila y procesa los datos sin formato de AWS IoT en métricas legibles y casi en tiempo real. Estas estadísticas se registran durante un periodo de dos semanas, de forma que pueda acceder a información histórica y obtener una mejor perspectiva sobre el rendimiento de su aplicación web o servicio. Por defecto, AWS IoT Los datos de métricas se envían automáticamente a CloudWatch en intervalos de un minuto. Para obtener más información, consulte [Qué son Amazon CloudWatch](#), [Amazon CloudWatch Eventos](#) y [Amazon CloudWatch ¿Registros?](#) en la [Amazon CloudWatch Guía del usuario](#).

Uso de las métricas de AWS IoT

Las métricas mostradas por AWS IoT proporcionan información que puede analizar de diferentes maneras. Los siguientes casos de uso se basan en una situación en la que tiene diez objetos que se conectan a Internet una vez al día. Cada día:

- Diez objetos se conectan con AWS IoT casi al mismo tiempo.
- Cada objeto se suscribe a un filtro de temas y, a continuación, espera una hora antes de desconectarse. Durante este periodo, los objetos se comunican entre sí y obtienen más información sobre el estado del mundo.
- Cada objeto publica alguna percepción que tenga según los datos que acaba de encontrar utilizando `UpdateThingShadow`.
- Cada objeto se desconecta de AWS IoT.

Para ayudarle a empezar, estos temas exploran algunas de las preguntas que podría tener.

- [¿Cómo se me puede enviar una notificación si mis objetos no se conectan correctamente cada día? \(p. 434\)](#)

- ¿Cómo se me puede enviar una notificación si mis objetos no publican datos cada día? (p. 435)
- ¿Cómo se me puede enviar una notificación si las actualizaciones de la sombra de mi objeto se rechazan cada día? (p. 435)
- ¿Cómo puedo crear una CloudWatch alarm for Jobs? (p. 436)

Más información sobre CloudWatch alarmas y métricas de

- [Crear CloudWatch alarmas para monitorearAWS IoT \(p. 434\)](#)
- [Métricas y dimensiones de AWS IoT \(p. 437\)](#)

Crear CloudWatch alarmas para monitorearAWS IoT

Puede crear una CloudWatch alarma que envía un mensaje de Amazon SNS cuando la alarma cambia de estado. Una alarma vigila una métrica individual durante un periodo de tiempo que usted especifica. Cuando el valor de la métrica supera un umbral determinado en un número de períodos de tiempo, se realizan una o varias acciones. La acción puede ser una notificación que se envía a un tema de Amazon SNS o a una política de Auto Scaling. Las alarmas activan acciones únicamente para los cambios de estado prolongados. CloudWatch Las alarmas de no activan acciones simplemente por tener un estado determinado. Es necesario que el estado haya cambiado y se mantenga durante un número especificado de periodos.

En los temas siguientes se describen algunos ejemplos del uso de alarmas de CloudWatch.

- [¿Cómo se me puede enviar una notificación si mis objetos no se conectan correctamente cada día? \(p. 434\)](#)
- [¿Cómo se me puede enviar una notificación si mis objetos no publican datos cada día? \(p. 435\)](#)
- [¿Cómo se me puede enviar una notificación si las actualizaciones de la sombra de mi objeto se rechazan cada día? \(p. 435\)](#)
- [¿Cómo puedo crear una CloudWatch alarma para trabajos? \(p. 436\)](#)

Puede ver todas las métricas de CloudWatch las alarmas pueden monitorizar en[Métricas y dimensiones de AWS IoT \(p. 437\)](#).

¿Cómo se me puede enviar una notificación si mis objetos no se conectan correctamente cada día?

1. Cree un tema de Amazon SNS denominado`things-not-connecting-successfully` registre su nombre de recurso de Amazon (ARN). Este procedimiento se referirá al ARN de su tema como ***sns-topic-arn***.

Para obtener más información sobre cómo crear una notificación de Amazon SNS, consulte[Introducción a Amazon SNS](#).

2. Cree la alarma.

```
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. Pruebe la alarma.

```
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. Compruebe que la alarma aparece en su[Consola de CloudWatch](#).

¿Cómo se me puede enviar una notificación si mis objetos no publican datos cada día?

1. Cree un tema de Amazon SNS denominado*things-not-publishing-data*y registre su nombre de recurso de Amazon (ARN). Este procedimiento se referirá al ARN de su tema como *sns-topic-arn*.

Para obtener más información sobre cómo crear una notificación de Amazon SNS, consulte[Introducción a Amazon SNS](#).

2. Cree la alarma.

```
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. Pruebe la alarma.

```
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. Compruebe que la alarma aparece en su[Consola de CloudWatch](#).

¿Cómo se me puede enviar una notificación si las actualizaciones de la sombra de mi objeto se rechazan cada día?

1. Cree un tema de Amazon SNS denominado*things-shadow-updates-rejected*y registre su nombre de recurso de Amazon (ARN). Este procedimiento se referirá al ARN de su tema como *sns-topic-arn*.

Para obtener más información sobre cómo crear una notificación de Amazon SNS, consulte[Introducción a Amazon SNS](#).

2. Cree la alarma.

```
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. Pruebe la alarma.

```
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. Compruebe que la alarma aparece en su[Consola de CloudWatch](#).

¿Cómo puedo crear una CloudWatch alarma para trabajos?

El servicio Jobs proporciona CloudWatch métricas para que pueda monitorizar sus trabajos. Puede crear alarmas de CloudWatch para monitorear cualquier [Métricas de trabajos \(p. 443\)](#).

El siguiente comando crea un CloudWatch alarma para supervisar el número total de ejecuciones de Job fallidas para trabajos [SampleOTAJob](#) y le notifica cuando más de 20 ejecuciones de trabajos han fracasado. La alarma supervisa la métrica de trabajos FailedJobExecutionTotalCount comprobando el valor notificado cada 300 segundos. Se activa cuando un único valor notificado es mayor que 20, lo que significa que hubo más de 20 ejecuciones de trabajo fallidas desde que se inició el trabajo. Cuando la alarma se apaga, envía una notificación al tema de Amazon SNS proporcionado.

```
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-executions
```

El siguiente comando crea un CloudWatch alarma para supervisar el número de ejecuciones de Job fallidas para trabajos [SampleOTAJob](#) en un período determinado. A continuación, le notifica cuando más de cinco ejecuciones de un trabajo han fracasado durante ese período. La alarma supervisa la métrica de trabajos FailedJobExecutionCount comprobando el valor notificado cada 3600 segundos. Se activa cuando un único valor notificado es mayor que 5, lo que significa que hubo más de 5 ejecuciones de

trabajo fallidas en la última hora. Cuando la alarma se apaga, envía una notificación al tema de Amazon SNS proporcionado.

```
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-executions-per-hour
```

Métricas y dimensiones de AWS IoT

Cuando se interactúa con AWS IoT, el servicio envía las siguientes métricas y dimensiones a CloudWatch cada minuto. Puede utilizar los siguientes procedimientos para consultar las métricas de AWS IoT.

Para ver las métricas (consola de CloudWatch)

Las métricas se agrupan en primer lugar por el espacio de nombres de servicio y, a continuación, por las diversas combinaciones de dimensiones dentro de cada espacio de nombres.

1. Abra la [consola de CloudWatch](#).
2. En el panel de navegación, elija **Métricas** y luego seleccione **Todas las métricas**.
3. En el navegador **Navegar pestaña**, busquen **AWS IoT** para ver la lista de métricas.

Para ver las métricas (CLI)

- En el símbolo del sistema, ejecute el siguiente comando:

```
aws cloudwatch list-metrics --namespace "AWS/IoT"
```

CloudWatch muestra los siguientes grupos de métricas para AWS IoT:

- [Métricas de AWS IoT \(p. 438\)](#)
- [AWS IoT Core métricas del proveedor de credenciales \(p. 438\)](#)
- [Métricas de reglas \(p. 438\)](#)
- [Métricas de acciones de reglas \(p. 439\)](#)
- [Métricas específicas de acciones HTTP \(p. 439\)](#)
- [Métricas del agente de mensajes \(p. 440\)](#)
- [Métricas de sombras de dispositivos \(p. 442\)](#)
- [Métricas de trabajos \(p. 443\)](#)
- [Métricas de auditoría de Device Defender \(p. 445\)](#)
- [Métricas de detección de Device Defender \(p. 445\)](#)
- [Métricas de aprovisionamiento de dispositivos \(p. 445\)](#)

- [Métricas de indexación de flotas \(p. 447\)](#)
- [Dimensiones de las métricas de \(p. 448\)](#)

Métricas de AWS IoT

Métrica	Descripción
AddThingToDynamicThingGroupsFailed	Número de eventos de error asociados a la incorporación de un objeto en un grupo de objetos dinámico. La dimensión <code>DynamicThingGroupName</code> contiene el nombre de los grupos dinámicos que no pudieron agregar objetos correctamente.
NumLogBatchesFailedToPublishThrottled	El lote de eventos de registro único que no se pudieron publicar debido a errores de limitación controlada.
NumLogEventsFailedToPublishThrottled	El número de eventos de registro en el lote que no se pudieron publicar debido a errores de limitación controlada.

AWS IoT Core métricas del proveedor de credenciales

Métrica	Descripción
CredentialExchangeSuccess	El número de personas realizadas correctamente <code>AssumeRoleWithCertificates</code> solicitudes de AWS IoT Core proveedor de credenciales de.

Métricas de reglas

Métrica	Descripción
ParseError	El número de errores de análisis JSON que se produjeron en los mensajes publicados en un tema en el que hay una regla a la escucha. La dimensión <code>RuleName</code> contiene el nombre de la regla.
RuleMessageThrottled	El número de mensajes limitados por el motor de reglas por un comportamiento malintencionado o porque el número de mensajes supera el límite del motor de reglas. La dimensión <code>RuleName</code> contiene el nombre de la regla que activar.
RuleNotFound	No se ha podido encontrar la regla que activar. La dimensión <code>RuleName</code> contiene el nombre de la regla.
RulesExecuted	El número de reglas de AWS IoT ejecutadas.
TopicMatch	El número de mensajes entrantes publicados en un tema en el que hay una regla a la escucha. La dimensión <code>RuleName</code> contiene el nombre de la regla.

Métricas de acciones de reglas

Métrica	Descripción
Failure	El número de llamadas a una acción de regla que produjeron un error. La dimensión <code>RuleName</code> contiene el nombre de la regla que especifica la acción. La dimensión <code>ActionType</code> contiene el tipo de acción que se invocó.
Success	El número de llamadas correctas a una acción de regla. La dimensión <code>RuleName</code> contiene el nombre de la regla que especifica la acción. La dimensión <code>ActionType</code> contiene el tipo de acción que se invocó.
ErrorActionFailure	Número de acciones de error que han producido un error. La dimensión <code>RuleName</code> contiene el nombre de la regla que especifica la acción. La dimensión <code>ActionType</code> contiene el tipo de acción que se invocó.
ErrorActionSuccess	El número de acciones de error realizadas correctamente. La dimensión <code>RuleName</code> contiene el nombre de la regla que especifica la acción. La dimensión <code>ActionType</code> contiene el tipo de acción que se invocó.

Métricas específicas de acciones HTTP

Métrica	Descripción
HttpCode_Other	Se genera si el código de estado de la respuesta del servicio o aplicación web de salida no es 2xx, 4xx o 5xx.
HttpCode_4XX	Se genera si el código de estado de la respuesta del servicio o aplicación web de salida está comprendido en el intervalo 400 y 499.
HttpCode_5XX	Se genera si el código de estado de la respuesta del servicio o aplicación web de salida está comprendido en el intervalo 500 y 599.
HttpInvalidUrl	Se genera si una URL de punto de enlace, una vez reemplazadas las plantillas de sustitución, no comienza por <code>https://</code> .
HttpRequestTimeout	Se genera si el servicio o la aplicación web de salida no devuelve ninguna respuesta dentro del límite de tiempo de espera de solicitud. Para obtener más información, consulte Cuotas de servicio .
HttpUnknownHost	Se genera si la URL es válida, pero el servicio no existe o no está accesible.

Métricas del agente de mensajes

Note

Las métricas del agente de mensajes se muestran en la CloudWatch consola bajo **Métricas de protocolo**.

Métrica	Descripción
<code>Connect.AuthError</code>	El número de solicitudes de conexión que el agente de mensajes no pudo autorizar. La dimensión <code>Protocol</code> contiene el protocolo usado para enviar el mensaje CONNECT.
<code>Connect.ClientError</code>	El número de solicitudes de conexión rechazadas porque el mensaje MQTT no cumplía los requisitos definidos en AWS IoT Cuotas de (p. 1294) . La dimensión <code>Protocol</code> contiene el protocolo usado para enviar el mensaje CONNECT.
<code>Connect.ClientIDThrottle</code>	El número de solicitudes de conexión que se rechazaron porque el cliente superó el límite de solicitudes de conexión permitidas para un ID de cliente específico. La dimensión <code>Protocol</code> contiene el protocolo usado para enviar el mensaje CONNECT.
<code>Connect.ServerError</code>	El número de solicitudes de conexión que fracasaron porque se produjo un error interno. La dimensión <code>Protocol</code> contiene el protocolo usado para enviar el mensaje CONNECT.
<code>Connect.Success</code>	El número de conexiones realizadas correctamente al agente de mensajes. La dimensión <code>Protocol</code> contiene el protocolo usado para enviar el mensaje CONNECT.
<code>Connect.Throttle</code>	Número de solicitudes de conexión que se rechazaron porque la cuenta superó el límite permitido. La dimensión <code>Protocol</code> contiene el protocolo usado para enviar el mensaje CONNECT.
<code>Ping.Success</code>	El número de mensajes ping recibidos por el agente de mensajes. La dimensión <code>Protocol</code> contiene el protocolo usado para enviar el mensaje ping.
<code>PublishIn.AuthError</code>	El número de solicitudes de publicación que el agente de mensajes no pudo autorizar. La dimensión <code>Protocol</code> contiene el protocolo usado para publicar el mensaje.
<code>PublishIn.ClientError</code>	El número de solicitudes de publicación rechazadas por el agente de mensajes porque el mensaje no cumplía los requisitos definidos en AWS IoT Cuotas de (p. 1294) . La dimensión <code>Protocol</code> contiene el protocolo usado para publicar el mensaje.
<code>PublishIn.ServerError</code>	El número de solicitudes de publicación que el agente de mensajes no pudo procesar porque se produjo un error interno. La dimensión <code>Protocol</code> contiene el protocolo usado para enviar el mensaje PUBLISH.

Métrica	Descripción
PublishIn.Success	El número de solicitudes de publicación que el agente de mensajes procesó correctamente. La dimensión Protocol contiene el protocolo usado para enviar el mensaje PUBLISH.
PublishIn.Throttle	El número de solicitudes de publicación que se rechazaron porque el cliente superó el límite de mensajes entrantes permitidos. La dimensión Protocol contiene el protocolo usado para enviar el mensaje PUBLISH.
PublishOut.AuthError	El número de solicitudes de publicación realizadas por el agente de mensajes que AWS IoT no pudo autorizar. La dimensión Protocol contiene el protocolo usado para enviar el mensaje PUBLISH.
PublishOut.ClientError	El número de solicitudes de publicación realizadas por el agente de mensajes que se rechazaron porque el mensaje no cumplía los requisitos definidos en AWS IoT Cuotas de (p. 1294) . La dimensión Protocol contiene el protocolo usado para enviar el mensaje PUBLISH.
PublishOut.Success	El número de solicitudes de publicación realizadas correctamente por el agente de mensajes. La dimensión Protocol contiene el protocolo usado para enviar el mensaje PUBLISH.
PublishOut.Throttle	El número de solicitudes de publicación que se rechazaron porque el cliente superó el límite de mensajes salientes permitidos. La dimensión Protocol contiene el protocolo usado para enviar el mensaje PUBLISH.
PublishRetained.AuthError	El número de solicitudes de publicación con laRETAINConjunto de indicadores que el agente de mensajes no pudo autorizar. La dimensión Protocol contiene el protocolo usado para enviar el mensaje PUBLISH.
PublishRetained.ServerError	El número de solicitudes de publicación que el agente de mensajes no pudo procesar porque se produjo un error interno. La dimensión Protocol contiene el protocolo usado para enviar el mensaje PUBLISH.
PublishRetained.Success	El número de solicitudes de publicación con laRETAINconjunto de indicadores que el agente de mensajes procesó correctamente. La dimensión Protocol contiene el protocolo usado para enviar el mensaje PUBLISH.
PublishRetained.Throttle	El número de solicitudes de publicación con laRETAINSe rechazaron porque el cliente superó el límite de mensajes entrantes permitidos. La dimensión Protocol contiene el protocolo usado para enviar el mensaje PUBLISH.

Métrica	Descripción
Subscribe.AuthError	El número de solicitudes de suscripción realizadas por un cliente que no se pudieron autorizar. La dimensión <code>Protocol</code> contiene el protocolo usado para enviar el mensaje <code>SUBSCRIBE</code> .
Subscribe.ClientError	El número de solicitudes de suscripción que se rechazaron porque el mensaje <code>SUBSCRIBE</code> no cumplía los requisitos definidos en AWS IoT Cuotas de (p. 1294) . La dimensión <code>Protocol</code> contiene el protocolo usado para enviar el mensaje <code>SUBSCRIBE</code> .
Subscribe.ServerError	El número de solicitudes de suscripción que se rechazaron porque se produjo un error interno. La dimensión <code>Protocol</code> contiene el protocolo usado para enviar el mensaje <code>SUBSCRIBE</code> .
Subscribe.Success	El número de solicitudes de suscripción que el agente de mensajes procesó correctamente. La dimensión <code>Protocol</code> contiene el protocolo usado para enviar el mensaje <code>SUBSCRIBE</code> .
Subscribe.Throttle	El número de solicitudes de suscripción que se rechazaron porque el cliente superó el límite de solicitudes de suscripción permitidas. La dimensión <code>Protocol</code> contiene el protocolo usado para enviar el mensaje <code>SUBSCRIBE</code> .
Unsubscribe.ClientError	Número de solicitudes de cancelación de suscripción que se rechazaron porque el mensaje <code>UNSUBSCRIBE</code> no cumplía los requisitos definidos en AWS IoT Cuotas de (p. 1294) . La dimensión <code>Protocol</code> contiene el protocolo usado para enviar el mensaje <code>UNSUBSCRIBE</code> .
Unsubscribe.ServerError	El número de solicitudes de cancelación de suscripción que se rechazaron porque se produjo un error interno. La dimensión <code>Protocol</code> contiene el protocolo usado para enviar el mensaje <code>UNSUBSCRIBE</code> .
Unsubscribe.Success	El número de solicitudes de cancelación de suscripción que el agente de mensajes procesó correctamente. La dimensión <code>Protocol</code> contiene el protocolo usado para enviar el mensaje <code>UNSUBSCRIBE</code> .
Unsubscribe.Throttle	El número de solicitudes de cancelación de suscripción que se rechazaron porque el cliente superó el límite de solicitudes de cancelación de suscripción permitidas. La dimensión <code>Protocol</code> contiene el protocolo usado para enviar el mensaje <code>UNSUBSCRIBE</code> .

Métricas de sombras de dispositivos

Note

Las métricas de sombras de dispositivo se muestran en la CloudWatch consola bajo `Métricas de protocolo`.

Métrica	Descripción
DeleteThingShadow.Accepted	El número de solicitudes DeleteThingShadow procesadas correctamente. La dimensión Protocolo contiene el protocolo usado para realizar la solicitud.
GetThingShadow.Accepted	El número de solicitudes GetThingShadow procesadas correctamente. La dimensión Protocolo contiene el protocolo usado para realizar la solicitud.
ListThingShadow.Accepted	El número de solicitudes ListThingShadow procesadas correctamente. La dimensión Protocolo contiene el protocolo usado para realizar la solicitud.
UpdateThingShadow.Accepted	El número de solicitudes UpdateThingShadow procesadas correctamente. La dimensión Protocolo contiene el protocolo usado para realizar la solicitud.

Métricas de trabajos

Métrica	Descripción
CanceledJobExecutionCount	Número de ejecuciones de trabajo cuyo estado ha cambiado a CANCELED durante un periodo determinado por CloudWatch. (Para obtener más información acerca de CloudWatch métricas, consulte Métricas de Amazon CloudWatch .) La dimensión JobId contiene el ID del trabajo.
CanceledJobExecutionTotalCount	El número total de ejecuciones de trabajo cuyo estado es CANCELED para el trabajo especificado. La dimensión JobId contiene el ID del trabajo.
ClientErrorCount	El número de errores de cliente generados mientras se ejecuta el trabajo. La dimensión JobId contiene el ID del trabajo.
FailedJobExecutionCount	Número de ejecuciones de trabajo cuyo estado ha cambiado a FAILED durante un periodo determinado por CloudWatch. (Para obtener más información acerca de CloudWatch métricas, consulte Amazon CloudWatch Métricas .) La dimensión JobId contiene el ID del trabajo.
FailedJobExecutionTotalCount	El número total de ejecuciones de trabajo cuyo estado es FAILED para el trabajo especificado. La dimensión JobId contiene el ID del trabajo.
InProgressJobExecutionCount	Número de ejecuciones de trabajo cuyo estado ha cambiado a IN_PROGRESS durante un periodo determinado por CloudWatch. (Para obtener más información acerca de CloudWatch métricas, consulte Métricas de Amazon CloudWatch .) La dimensión JobId contiene el ID del trabajo.

Métrica	Descripción
InProgressJobExecutionTotalCount	El número total de ejecuciones de trabajo cuyo estado es IN_PROGRESS para el trabajo especificado. La dimensión JobId contiene el ID del trabajo.
RejectedJobExecutionTotalCount	El número total de ejecuciones de trabajo cuyo estado es REJECTED para el trabajo especificado. La dimensión JobId contiene el ID del trabajo.
RemovedJobExecutionTotalCount	El número total de ejecuciones de trabajo cuyo estado es REMOVED para el trabajo especificado. La dimensión JobId contiene el ID del trabajo.
QueuedJobExecutionCount	Número de ejecuciones de trabajo cuyo estado ha cambiado a QUEUED durante un periodo determinado por CloudWatch. (Para obtener más información acerca de CloudWatch métricas, consulte Amazon CloudWatch Métricas .) La dimensión JobId contiene el ID del trabajo.
QueuedJobExecutionTotalCount	El número total de ejecuciones de trabajo cuyo estado es QUEUED para el trabajo especificado. La dimensión JobId contiene el ID del trabajo.
RejectedJobExecutionCount	Número de ejecuciones de trabajo cuyo estado ha cambiado a REJECTED durante un periodo determinado por CloudWatch. (Para obtener más información acerca de CloudWatch métricas, consulte Métricas de Amazon CloudWatch .) La dimensión JobId contiene el ID del trabajo.
RemovedJobExecutionCount	Número de ejecuciones de trabajo cuyo estado ha cambiado a REMOVED durante un periodo determinado por CloudWatch. (Para obtener más información acerca de CloudWatch métricas, consulte Métricas de Amazon CloudWatch .) La dimensión JobId contiene el ID del trabajo.
ServerErrorCount	El número de errores de servidor generados mientras se ejecuta el trabajo. La dimensión JobId contiene el ID del trabajo.
SucceededJobExecutionCount	Número de ejecuciones de trabajo cuyo estado ha cambiado a SUCCESS durante un periodo determinado por CloudWatch. (Para obtener más información acerca de CloudWatch métricas, consulte Métricas de Amazon CloudWatch .) La dimensión JobId contiene el ID del trabajo.
SucceededJobExecutionTotalCount	El número total de ejecuciones de trabajo cuyo estado es SUCCESS para el trabajo especificado. La dimensión JobId contiene el ID del trabajo.

Métricas de auditoría de Device Defender

Métrica	Descripción
NonCompliantResources	Número de recursos que se ha comprobado que no cumplen los requisitos de una comprobación. El sistema notifica el número de recursos no conformes en cada comprobación de cada auditoría realizada.
ResourcesEvaluated	Número de recursos cuya conformidad se evaluó. El sistema notifica el número de recursos que se evaluaron en cada comprobación de cada auditoría realizada.

Métricas de detección de Device Defender

Métrica	Descripción
Violations	El número de nuevas infracciones de los comportamientos del perfil de seguridad que se han encontrado desde la última vez que se realizó una evaluación. El sistema comunica el número de infracciones nuevas de la cuenta, de un perfil de seguridad específico y de un comportamiento concreto de un perfil de seguridad determinado.
ViolationsCleared	El número de infracciones de los comportamientos del perfil de seguridad que se han resuelto desde la última vez que se realizó una evaluación. El sistema comunica el número de infracciones resueltas de la cuenta, para un perfil de seguridad específico y para un comportamiento concreto de un perfil de seguridad determinado.
ViolationsInvalidated	El número de infracciones de los comportamientos del perfil de seguridad de las que ya no está disponible la información desde la última vez que se realizó una evaluación (debido a que el dispositivo de informe dejó de realizar informes o a que ya no se monitoriza por algún motivo). El sistema comunica el número de infracciones invalidadas de toda la cuenta, de un perfil de seguridad específico y de un comportamiento concreto de un perfil de seguridad determinado.

Métricas de aprovisionamiento de dispositivos

AWS IoT Métricas de aprovisionamiento de flotas

Métrica	Descripción
ApproximateNumberOfThingsRegistered	El número de objetos que se han registrado en Fleet Provisioning. Si bien el recuento es en general preciso, la arquitectura distribuida de AWS IoT dificulta mantener un recuento preciso de las cosas registradas.

Métrica	Descripción
	<p>La estadística que se utiliza para esta métrica es:</p> <ul style="list-style-type: none"> • Max (Máximo)para informar del número total de cosas que se han registrado. Para obtener un recuento de las cosas registradas durante la ventana de agregación de CloudWatch, consulte laRegisterThingFailedMétrica de. <p>Dimensiones: ID de certificado de reclamación (p. 448)</p>
CreateKeysAndCertificateFailed	<p>Número de errores que se produjeron por las llamadas a laCreateKeysAndCertificateAPI MQTT.</p> <p>La métrica se emite en casos de éxito (valor = 0) y Fallo (valor = 1). Esta métrica se puede utilizar para realizar un seguimiento del número de certificados creados y registrados durante las ventanas de agregación compatibles con Cloudwatch, como 5 minutos o 1 hora.</p> <p>Las estadísticas disponibles para esta métrica son:</p> <ul style="list-style-type: none"> • Sum (Suma)para informar del número de llamadas fallidas. • Recuento de ejemplopara informar del número total de llamadas correctas y fallidas.
CreateCertificateFromCsrFailed	<p>Número de errores que se produjeron por las llamadas a laCreateCertificateFromCsrAPI MQTT.</p> <p>La métrica se emite en casos de éxito (valor = 0) y Fallo (valor = 1). Esta métrica se puede utilizar para realizar un seguimiento del número de cosas registradas durante las ventanas de agregación compatibles con Cloudwatch, como 5 minutos o 1 hora.</p> <p>Las estadísticas disponibles para esta métrica son:</p> <ul style="list-style-type: none"> • Sum (Suma)para informar del número de llamadas fallidas. • Recuento de ejemplopara informar del número total de llamadas correctas y fallidas.

Métrica	Descripción
RegisterThingFailed	<p>Número de errores que se produjeron por las llamadas a la RegisterThingAPI MQTT.</p> <p>La métrica se emite en casos de éxito (valor = 0) y Fallo (valor = 1). Esta métrica se puede utilizar para realizar un seguimiento del número de cosas registradas durante las ventanas de agregación compatibles con Cloudwatch, como 5 minutos o 1 hora. Para obtener información sobre el número total de elementos registrados, consulte la ApproximateNumberOfThingsRegistered Métrica de.</p> <p>Las estadísticas disponibles para esta métrica son:</p> <ul style="list-style-type: none"> • Sum (Suma) para informar del número de llamadas fallidas. • Recuento de ejemplos para informar del número total de llamadas correctas y fallidas. <p>Dimensiones: TemplateName (p. 448)</p>

Métricas de aprovisionamiento justo a tiempo

Métrica	Descripción
ProvisionThing.ClientError	Número de veces que un dispositivo no pudo aprovisionar debido a un error del cliente. Por ejemplo, la política especificada en la plantilla no existía.
ProvisionThing.ServerError	Número de veces que un dispositivo no pudo aprovisionar debido a un error de servidor. Los clientes pueden volver a intentar aprovisionar el dispositivo después de esperar y pueden ponerse en contacto con AWS IoT si el problema sigue siendo el mismo.
ProvisionThing.Success	El número de veces que un dispositivo se aprovisionó correctamente.

Métricas de indexación de flotas

AWS IoT Métricas de indexación de flota

Métrica	Descripción
DeviceDefenderThingViolationsEventsSizeLimitExceeded	El tamaño total de los datos de una cosa procesada por la indexación de flotas está limitado a 32 KB. Cuando se incumple este límite para una cosa debido a un evento de infracciones de Device Defender, el DeviceDefenderThingViolationsEventSizeLimitExceeded emitirá el tipo de evento.
NamedShadowEventSizeLimitExceeded	El tamaño total de los datos de una cosa procesada por la indexación de flotas está limitado a 32 KB.

Métrica	Descripción
	Cuando se infringe este límite para una cosa debido a un evento de sombra con nombre, elNamedShadowEventSizeLimitExceeded se emitirá el tipo de evento.
NamedShadowCountForDynamicGroupQuery	Se imprime el límite de 5 sombras con nombre por cosa para términos de consulta que no son específicos del origen de datos en grupos de cosas dinámicos. Cuando se incumple este límite para una cosa, elNamedShadowCountForDynamicGroupQueryLimitExceeded se emitirá el tipo de evento.

Dimensiones de las métricas de

Las métricas utilizan el espacio de nombres y proporcionan métricas para las siguientes dimensiones.

Dimensión	Descripción
ActionType	El tipo de acción (p. 479) especificado por la regla que activó la solicitud.
BehaviorName	El nombre del comportamiento del perfil de seguridad de Device Defender Detect que se está monitorizando.
ClaimCertificateId	La certificateId de la reclamación utilizada para suministrar los dispositivos.
CheckName	El nombre de la comprobación de auditoría Device Defender cuyos resultados se están monitoreando.
JobId	El ID del trabajo cuyo progreso o tasa de éxito/error para la conexión de mensajes se está monitorizando.
Protocol	El protocolo utilizado para realizar la solicitud. Los valores válidos son: MQTT o HTTP
RuleName	El nombre de la regla activada por la solicitud.
ScheduledAuditName	El nombre de la auditoría programada de Device Defender cuyos resultados de comprobación se están monitoreando. Tiene el valor OnDemand si los resultados registrados corresponden a una auditoría que se realizó bajo demanda.
SecurityProfileName	El nombre del perfil de seguridad de detección de Device Defender cuyos comportamientos se están monitoreando.
TemplateName	El nombre de la plantilla de aprovisionamiento.

MonitorAWS IoT con CloudWatch Registros

Cuando el [registro de AWS IoT está habilitado \(p. 427\)](#), AWS IoT envía eventos de progreso acerca de cada mensaje a medida que este pasa desde los dispositivos al agente de mensajes y al motor de reglas. En el navegador[Consola de CloudWatch](#), CloudWatch Los registros aparecen en un grupo de registros denominadoAWSLogs.

Para obtener más información acerca de CloudWatch Registros, consulte[Registros de CloudWatch](#). Para obtener información sobre los admitidosAWS IoT CloudWatch Registros, consulte[CloudWatchAWS IoT Entradas de registro de en \(p. 449\)](#).

Visualización deAWS IoT inicios de sesión en el CloudWatch consola

Note

LaAWSLogsV2el grupo de registros no está visible en el CloudWatch consola hasta:

- Has habilitado el inicio de sesiónAWS IoT. Para obtener más información sobre cómo habilitar el inicio de sesiónAWS IoT, consulte[Configuración de registros de AWS IoT \(p. 427\)](#)
- Algunas entradas de registro han sido escritas porAWS IoT operaciones de.

Para ver losAWS IoT inicios de sesión en el CloudWatch consola

1. Vaya a<https://console.aws.amazon.com/cloudwatch/>. En el panel de navegación, elija Log groups (Grupos de registro).
2. En el cuadro Filter (Filtro), escriba **AWSLogsV2** y, a continuación, pulse Intro.
3. Haga doble clic en el grupo de registros AWSLogsV2.
4. ElegirBuscar en todo. Se genera una lista completa de registros de AWS IoT para la cuenta.
5. Elija el icono de ampliar para analizar un flujo individual.

También puede escribir una consulta en el cuadro de texto Filter events (Filtrar eventos). Aquí tiene algunas consultas interesantes que probar:

- { \$.logLevel = "INFO" }
- Busque todos los registros que tengan un nivel de registro de INFO.
- { \$.status = "Success" }
- Busque todos los registros que tengan un estado de Success.
- { \$.status = "Success" && \$.eventType = "GetThingShadow" }
- Busque todos los registros que tengan un estado de Success y un tipo de evento de GetThingShadow.

Para obtener más información acerca de la creación de expresiones de filtro, consulte[Consultas de CloudWatch Logs](#).

CloudWatchAWS IoT Entradas de registro de en

Cada componente de AWS IoT genera sus propias entradas de registro. Cada entrada de registro tiene un eventType que especifica la operación que provocó que se genere la entrada de registro. En esta sección

se describen las entradas de registro generadas por los siguientes AWS IoT componentes. Para obtener información sobre AWS IoT Core para la supervisión de LoRaWAN, consulte [Vista CloudWatch AWS IoT Wireless Entradas de registro de en \(p. 1232\)](#).

Temas

- [Entradas de registro del agente de mensajes \(p. 450\)](#)
- [Entradas de registro de sombre de dispositivo \(p. 455\)](#)
- [Entradas del registro del motor de reglas \(p. 457\)](#)
- [Entradas del registro de Job \(p. 461\)](#)
- [Entradas de registro de aprovisionamiento de dispositivos \(p. 465\)](#)
- [Entradas de registro de grupo de objetos dinámicos \(p. 466\)](#)
- [Entradas de los registros de indexación de \(p. 467\)](#)
- [Común CloudWatch Atributos registros \(p. 469\)](#)

Entradas de registro del agente de mensajes

El agente de mensajes de AWS IoT genera entradas de registro para los siguientes eventos:

Temas

- [Entrada de registro Connect \(p. 450\)](#)
- [Entrada de registro Disconnect \(p. 451\)](#)
- [Entrada de registro GetRetainedMessage \(p. 452\)](#)
- [Entrada de registro de lista de mensajes retenidos \(p. 452\)](#)
- [Entrada de registro Publish-In \(p. 453\)](#)
- [Entrada de registro Publish-Out \(p. 453\)](#)
- [Entrada de registro Subscribe \(p. 454\)](#)

Entrada de registro Connect

El agente de mensajes de AWS IoT genera una entrada de registro con un eventType de Connect cuando se conecta un cliente MQTT.

Ejemplo de entrada de registro Connect

```
{  
    "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  
}
```

Además de [Común CloudWatch Atributos registros \(p. 469\)](#), las entradas de registro Connect contienen los siguientes atributos:

clientId

El ID del cliente que realiza la solicitud.

principalId

El ID de la entidad principal que realiza la solicitud.

protocolo

El protocolo utilizado cuando se realiza la solicitud. Los valores válidos son MQTT o HTTP.

sourceIp

La dirección IP en la que se originó la solicitud.

sourcePort

El puerto en el que se originó la solicitud.

Entrada de registro Disconnect

El agente de mensajes de AWS IoT genera una entrada de registro con un eventType de Disconnect cuando se desconecta un cliente MQTT.

Ejemplo de entrada de registro Disconnect

```
{  
    "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"  
}
```

Además de [Común CloudWatch Atributos registros \(p. 469\)](#), las entradas de registro Disconnect contienen los siguientes atributos:

clientId

El ID del cliente que realiza la solicitud.

principalId

El ID de la entidad principal que realiza la solicitud.

protocolo

El protocolo utilizado cuando se realiza la solicitud. Los valores válidos son MQTT o HTTP.

sourceIp

La dirección IP en la que se originó la solicitud.

sourcePort

El puerto en el que se originó la solicitud.

disconnectReason

La razón por la que el cliente se está desconectando.

Entrada de registro GetRetainedMessage

La AWS IoT agente de mensajes genera una entrada de registro con un evento tipo `GetRetainedMessage` cuando `GetRetainedMessages` se invoca a.

Ejemplo de entrada de registro 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"  
}
```

Además de [Común CloudWatch Atributos registros \(p. 469\)](#), las entradas de registro `GetRetainedMessage` contienen los siguientes atributos:

`lastModifiedDate`

La fecha y hora de época, en milisegundos, en que el mensaje conservado fue almacenado por AWS IoT.

`protocol`

El protocolo utilizado cuando se realiza la solicitud. Valor válido: `HTTP`.

`qos`

El nivel de calidad de servicio (QoS) utilizado en la solicitud de publicación. Los valores válidos son 0 o 1.

`topicName`

El nombre del tema suscrito.

Entrada de registro de lista de mensajes retenidos

La AWS IoT agente de mensajes genera una entrada de registro con un evento tipo `ListRetainedMessage` cuando `ListRetainedMessages` se invoca a.

Ejemplo de entrada de registro 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"  
}
```

Además de la [Común CloudWatch Atributos registros \(p. 469\)](#), `ListRetainedMessages` las entradas de registro contienen el siguiente atributo:

protocolo

El protocolo utilizado cuando se realiza la solicitud. Valor válido:HTTP.

Entrada de registro Publish-In

Cuando el agente de mensajes de AWS IoT recibe un mensaje MQTT, genera una entrada de registro con un eventType de Publish-In.

Ejemplo de entrada de registro Publish-In

```
{  
    "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  
}
```

Además de Común CloudWatch Atributos registros (p. 469), las entradas de registro Publish-In contienen los siguientes atributos:

clientId

El ID del cliente que realiza la solicitud.

principalId

El ID de la entidad principal que realiza la solicitud.

protocolo

El protocolo utilizado cuando se realiza la solicitud. Los valores válidos son MQTT o HTTP.

sourceIp

La dirección IP en la que se originó la solicitud.

sourcePort

El puerto en el que se originó la solicitud.

topicName

El nombre del tema suscrito.

Entrada de registro Publish-Out

Cuando el agente de mensajes publica un mensaje MQTT, genera una entrada de registro con un eventType de Publish-Out.

Ejemplo de entrada de registro Publish-Out

```
{
```

```
    "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
}
```

Además de [Común CloudWatch Atributos registros \(p. 469\)](#), las entradas de registro Publish-Out contienen los siguientes atributos:

clientId

El ID del cliente suscrito que recibe mensajes sobre ese tema MQTT.

principalId

El ID de la entidad principal que realiza la solicitud.

protocolo

El protocolo utilizado cuando se realiza la solicitud. Los valores válidos son MQTT o HTTP.

sourceIp

La dirección IP en la que se originó la solicitud.

sourcePort

El puerto en el que se originó la solicitud.

topicName

El nombre del tema suscrito.

Entrada de registro Subscribe

El agente de mensajes de AWS IoT genera una entrada de registro con un eventType de Subscribe cuando un cliente MQTT se suscribe a un tema.

Ejemplo de entrada de registro Subscribe

```
{
    "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
}
```

Además de Común CloudWatch Atributos registros (p. 469), las entradas de registro Subscribe contienen los siguientes atributos:

clientId

El ID del cliente que realiza la solicitud.

principalId

El ID de la entidad principal que realiza la solicitud.

protocolo

El protocolo utilizado cuando se realiza la solicitud. Los valores válidos son MQTT o HTTP.

sourceIp

La dirección IP en la que se originó la solicitud.

sourcePort

El puerto en el que se originó la solicitud.

topicName

El nombre del tema suscrito.

Entradas de registro de sombre de dispositivo

El servicio de sombra de dispositivo de AWS IoT genera entradas de registro para los siguientes eventos:

Temas

- [Entrada de registro DeleteThingShadow \(p. 455\)](#)
- [Entrada de registro GetThingShadow \(p. 456\)](#)
- [Entrada de registro UpdateThingShadow \(p. 456\)](#)

Entrada de registro DeleteThingShadow

El servicio de sombra de dispositivo genera una entrada de registro con un eventType de DeleteThingShadow cuando se recibe una solicitud de eliminación de la sombra de un dispositivo.

Ejemplo de entrada de registro 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"  
}
```

Además de Común CloudWatch Atributos registros (p. 469), las entradas de registro DeleteThingShadow contienen los siguientes atributos:

deviceShadowName

Nombre de la sombra que se va a actualizar.

protocolo

El protocolo utilizado cuando se realiza la solicitud. Los valores válidos son MQTT o HTTP.
topicName

El nombre del tema en el que se publicó la solicitud.

Entrada de registro GetThingShadow

El servicio de sombra de dispositivo genera una entrada de registro con un eventType de GetThingShadow cuando se recibe una solicitud de obtención para una sombra.

Ejemplo de entrada de registro 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"  
}
```

Además de Común CloudWatch Atributos registros (p. 469), las entradas de registro GetThingShadow contienen los siguientes atributos:

deviceShadowName

El nombre de la sombra solicitada.

protocolo

El protocolo utilizado cuando se realiza la solicitud. Los valores válidos son MQTT o HTTP.
topicName

El nombre del tema en el que se publicó la solicitud.

Entrada de registro UpdateThingShadow

El servicio de sombra de dispositivo genera una entrada de registro con un eventType de UpdateThingShadow cuando se recibe una solicitud de actualización de la sombra de un dispositivo.

Ejemplo de entrada de registro 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"  
}
```

Además de Común CloudWatch Atributos registros (p. 469), las entradas de registro UpdateThingShadow contienen los siguientes atributos:

deviceShadowName

Nombre de la sombra que se va a actualizar.

protocolo

El protocolo utilizado cuando se realiza la solicitud. Los valores válidos son MQTT o HTTP.

topicName

El nombre del tema en el que se publicó la solicitud.

Entradas del registro del motor de reglas

El motor de reglas de AWS IoT genera registros para los siguientes eventos:

Temas

- [Entrada de registro FunctionExecution \(p. 457\)](#)
- [Entrada de registro de RuleExecution \(p. 458\)](#)
- [Entrada de registro RuleMatch \(p. 459\)](#)
- [Entrada de registro RuleMessageThrottled \(p. 459\)](#)
- [Entrada de registro RuleNotFound \(p. 460\)](#)
- [Entrada de registro StartingRuleExecution \(p. 461\)](#)

Entrada de registro FunctionExecution

El motor de reglas genera una entrada de registro con un eventType de FunctionExecution cuando la consulta SQL de una regla llama a una función externa. Se llama a una función externa cuando una acción de la regla realiza una solicitud HTTP a AWS IoT u otro servicio web (por ejemplo, llamar a get_thing_shadow o machinelearning_predict).

Ejemplo de entrada de registro de 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"  
}
```

Además de Común CloudWatch Atributos registros (p. 469), las entradas de registro FunctionExecution contienen los siguientes atributos:

clientId

N/A para registros FunctionExecution.

principalId

El ID de la entidad principal que realiza la solicitud.

recursos

Un conjunto de recursos utilizados por las acciones de la regla.

ruleName

El nombre de la regla que coincide.

topicName

El nombre del tema suscrito.

Entrada de registro de RuleExecution

Cuando el motor de reglas de AWS IoT activa la acción de una regla, genera un registro de RuleExecution.

Ejemplo de entrada de registro de 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"  
}
```

Además de [Común CloudWatch Atributos registros \(p. 469\)](#), las entradas de registro RuleExecution contienen los siguientes atributos:

clientId

El ID del cliente que realiza la solicitud.

principalId

El ID de la entidad principal que realiza la solicitud.

recursos

Un conjunto de recursos utilizados por las acciones de la regla.

ruleAction

El nombre de la acción activada.

ruleName

El nombre de la regla que coincide.

topicName

El nombre del tema suscrito.

Entrada de registro RuleMatch

El motor de reglas de AWS IoT genera una entrada de registro con un eventType de RuleMatch cuando el agente de mensajes recibe un mensaje que coincide con una regla.

Ejemplo de entrada de registro 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"  
}
```

Además de [Común CloudWatch Atributos registros \(p. 469\)](#), las entradas de registro RuleMatch contienen los siguientes atributos:

clientId

El ID del cliente que realiza la solicitud.

principalId

El ID de la entidad principal que realiza la solicitud.

ruleName

El nombre de la regla que coincide.

topicName

El nombre del tema suscrito.

Entrada de registro RuleMessageThrottled

Cuando se limita un mensaje, el motor de reglas de AWS IoT genera una entrada de registro con un eventType de RuleMessageThrottled.

Ejemplo de entrada de registro 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"  
}
```

Además de [Común CloudWatch Atributos registros \(p. 469\)](#), las entradas de registro RuleMessageThrottled contienen los siguientes atributos:

clientId

El ID del cliente que realiza la solicitud.

details

Una breve explicación del error.

principalId

El ID de la entidad principal que realiza la solicitud.

reason

La cadena "RuleMessageThrottled".

ruleName

El nombre de la regla que se debe activar.

topicName

El nombre del tema publicado.

Entrada de registro RuleNotFound

Cuando el motor de reglas de AWS IoT no puede encontrar una regla con un nombre concreto, genera una entrada de registro con un eventType de RuleNotFound.

Ejemplo de entrada de registro 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": "145179c40e2219e18a090d896a5340b74cf97a39641beec2fc3eeafc5a932167",  
    "reason": "RuleNotFound",  
    "details": "Rule example_rule not found"  
}
```

Además de [Común CloudWatch Atributos registros \(p. 469\)](#), las entradas de registro RuleNotFound contienen los siguientes atributos:

clientId

El ID del cliente que realiza la solicitud.

details

Una breve explicación del error.

principalId

El ID de la entidad principal que realiza la solicitud.

reason

La cadena "RuleNotFound".

ruleName

El nombre de la regla que no se pudo encontrar.

topicName

El nombre del tema publicado.

Entrada de registro StartingRuleExecution

Cuando el motor de reglas de AWS IoT empieza a activar la acción de una regla, genera una entrada de registro con un eventType de StartingRuleExecution.

Ejemplo de entrada de registro 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"  
}
```

Además de [Común CloudWatch Atributos registros \(p. 469\)](#), las entradas de registro rule- contienen los siguientes atributos:

clientId

El ID del cliente que realiza la solicitud.

principalId

El ID de la entidad principal que realiza la solicitud.

ruleAction

El nombre de la acción activada.

ruleName

El nombre de la regla que coincide.

topicName

El nombre del tema suscrito.

Entradas del registro de Job

El servicio Job de AWS IoT genera entradas de registro para los siguientes eventos. Las entradas de registro se generan cuando se recibe una solicitud de MQTT o HTTP procedente del dispositivo.

Temas

- [Entrada de registro DescribeJobExecution \(p. 462\)](#)
- [Entrada de registro GetPendingJobExecution \(p. 462\)](#)
- [Entrada de registro ReportFinalJobExecutionCount \(p. 463\)](#)

- Entrada de registro StartNextPendingJobExecution (p. 464)
- Entrada de registro UpdateJobExecution (p. 464)

Entrada de registro DescribeJobExecution

El servicio Jobs de AWS IoT genera una entrada de registro con un eventType de `DescribeJobExecution` cuando el servicio recibe una solicitud para describir la ejecución de un trabajo.

Ejemplo de entrada de registro 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."  
}
```

Además de [Común CloudWatch Atributos registros \(p. 469\)](#), las entradas de registro `GetJobExecution` contienen los siguientes atributos:

clientId

El ID del cliente que realiza la solicitud.

clientToken

Identificador único con distinción entre mayúsculas y minúsculas único para garantizar la idempotencia de la solicitud. Para obtener más información, consulte [How to Ensure Idempotency](#).

details

Información adicional del servicio Jobs.

jobId

El ID de trabajo para la ejecución de trabajos.

protocolo

El protocolo utilizado cuando se realiza la solicitud. Los valores válidos son `MQTT` o `HTTP`.

topicName

El tema utilizado para realizar la solicitud.

Entrada de registro GetPendingJobExecution

El servicio de Jobs de AWS IoT genera una entrada de registro con un eventType de `GetPendingJobExecution` cuando el servicio recibe una solicitud de ejecución de un trabajo.

Ejemplo de entrada de registro 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."
}
```

Además de [Común CloudWatch Atributos registros \(p. 469\)](#), las entradas de registro GetPendingJobExecution contienen los siguientes atributos:

clientId

El ID del cliente que realiza la solicitud.

clientToken

Identificador único con distinción entre mayúsculas y minúsculas que permite garantizar la idempotencia de la solicitud. Para obtener más información, consulte [How to Ensure Idempotency](#).

details

Información adicional del servicio Jobs.

protocolo

El protocolo utilizado cuando se realiza la solicitud. Los valores válidos son MQTT o HTTP.

topicName

El nombre del tema suscrito.

Entrada de registro ReportFinalJobExecutionCount

La AWS IoT servicio Jobs genera una entrada de registro con un entryType de ReportFinalJobExecutionCount cuando se completa un trabajo.

Ejemplo de entrada de registro 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"
}
```

Además de [Común CloudWatch Atributos registros \(p. 469\)](#), las entradas de registro ReportFinalJobExecutionCount contienen los siguientes atributos:

details

Información adicional del servicio Jobs.

jobId

El ID de trabajo para la ejecución de trabajos.

Entrada de registro StartNextPendingJobExecution

Cuando recibe una solicitud para iniciar la siguiente ejecución de trabajo pendiente, el servicio Jobs de AWS IoT genera una entrada de registro con un eventType de StartNextPendingJobExecution.

Ejemplo de entrada de registro 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."  
}
```

Además de [Común CloudWatch Atributos registros \(p. 469\)](#), las entradas de registro StartNextPendingJobExecution contienen los siguientes atributos:

clientId

El ID del cliente que realiza la solicitud.

clientToken

Identificador único con distinción entre mayúsculas y minúsculas que permite garantizar la idempotencia de la solicitud. Para obtener más información, consulte [How to Ensure Idempotency](#).

details

Información adicional del servicio Jobs.

protocolo

El protocolo utilizado cuando se realiza la solicitud. Los valores válidos son MQTT o HTTP.

topicName

El tema utilizado para realizar la solicitud.

Entrada de registro UpdateJobExecution

El servicio Jobs de AWS IoT genera una entrada de registro con un eventType de UpdateJobExecution cuando el servicio recibe una solicitud para actualizar una ejecución de un trabajo.

Ejemplo de entrada de registro 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."  
}
```

Además de Común CloudWatch Atributos registros (p. 469), las entradas de registro UpdateJobExecution contienen los siguientes atributos:

clientId

El ID del cliente que realiza la solicitud.

clientToken

Identificador único con distinción entre mayúsculas y minúsculas que permite garantizar la idempotencia de la solicitud. Para obtener más información, consulte How to Ensure Idempotency.

details

Información adicional del servicio Jobs.

jobId

El ID de trabajo para la ejecución de trabajos.

protocolo

El protocolo utilizado cuando se realiza la solicitud. Los valores válidos son MQTT o HTTP.

topicName

El tema utilizado para realizar la solicitud.

versionNumber

La versión de la ejecución de trabajos.

Entradas de registro de aprovisionamiento de dispositivos

El servicio Aprovisionamiento de dispositivos de AWS IoT genera registros para los siguientes eventos:

Temas

- Entrada de registro GetDeviceCredentials (p. 465)
- Entrada de registro de ProvisionDevice (p. 466)

Entrada de registro GetDeviceCredentials

El servicio de aprovisionamiento de dispositivos de AWS IoT genera una entrada de registro con un eventType de GetDeviceCredential cuando un cliente llama a GetDeviceCredential.

Ejemplo de entrada de registro 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."  
}
```

Además de Común CloudWatch Atributos registros (p. 469), las entradas de registro GetDeviceCredentials contienen los siguientes atributos:

details

Una breve explicación del error.

deviceCertificateId

El ID del certificado del dispositivo.

Entrada de registro de ProvisionDevice

El servicio de aprovisionamiento de dispositivos de AWS IoT genera una entrada de registro con un eventType de ProvisionDevice cuando un cliente llama a ProvisionDevice.

Ejemplo de entrada de registro de 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."  
}
```

Además de Común CloudWatch Atributos registros (p. 469), las entradas de registro ProvisionDevice contienen los siguientes atributos:

details

Una breve explicación del error.

deviceCertificateId

El ID del certificado del dispositivo.

provisioningTemplateName

El nombre de la plantilla de aprovisionamiento.

Entradas de registro de grupo de objetos dinámicos

Los grupos de objetos dinámicos de AWS IoT generan registros para el siguiente evento.

Temas

- Entrada de registro AddThingToDynamicThingGroupsFailed (p. 466)

Entrada de registro AddThingToDynamicThingGroupsFailed

Cuando AWS IoT no es capaz de agregar un objeto a los grupos dinámicos especificados, genera una entrada de registro con un eventType de AddThingToDynamicThingGroupsFailed. Esto ocurre cuando un objeto cumplía los criterios para estar en el grupo de objetos dinámico, pero no se pudo agregar a este grupo o se eliminó de él. Esto puede suceder por los motivos siguientes:

- El objeto ya es miembro del número máximo de grupos.
- Se utilizó la opción `--override-dynamic-groups` para agregar el objeto a un grupo de objetos estático. Se eliminó de un grupo de objetos dinámico para hacerlo posible.

Para obtener más información, consulte este artículo sobre las [limitaciones y conflictos de los grupos de objetos dinámicos \(p. 285\)](#).

Ejemplo de entrada de registro AddThingToDynamicThingGroupsFailed

En este ejemplo, se muestra la entrada de registro de un error `AddThingToDynamicThingGroupsFailed`. En este ejemplo, `TestThing` cumplía los criterios para estar en los grupos de objetos dinámicos que se indican en `dynamicThingGroupNames`, pero no pudo agregarse a esos grupos dinámicos, tal y como se describe en `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."  
}
```

Además de [Común CloudWatch Atributos registros \(p. 469\)](#), las entradas de registro `AddThingToDynamicThingGroupsFailed` contienen los siguientes atributos:

`dynamicThingGroupNames`

Matriz de los grupos de objetos dinámicos a los que no pudo agregarse el objeto.

`reason`

Razón por la cual el objeto no pudo agregarse a los grupos dinámicos.

`thingName`

Nombre del objeto que no pudo agregarse a un grupo de objetos dinámico.

Entradas de los registros de indexación de

AWS IoT La indexación de flotas de genera entradas de registro para los siguientes eventos.

Temas

- [Entrada de registro DeviceDefender ThingViolationSeveElimit Se ha superado la entrada de registro \(p. 468\)](#)
- [Entrada de registro NameShadowEvents SizeLimitExcedió la entrada de registro \(p. 468\)](#)
- [Entrada de registro NameShadowCount para DynamicGroupQueryLimiTExcedió la entrada de registro \(p. 468\)](#)

Entrada de registro DeviceDefender ThingViolationSeveElimit Se ha superado la entrada de registro

El tamaño total de los datos de una cosa procesada por la indexación de flotas está limitado a 32 KB. Cuando se incumple este límite para una cosa debido a un evento de infracciones de Device Defender, el DeviceDefenderThingViolationsEventSizeLimitExceeded se emitirá el tipo de evento.

Ejemplo de entrada de registro DeviceDefenderThingViolationSeventSizeElimitExceed

En este ejemplo, se muestra la entrada de registro de unDeviceDefenderThingViolationsEventSizeLimitExceeded. En este ejemplo, la cosa llamada TestThing tiene datos de infracciones que se estaban llenando, pero se omitió la indexación o filtración de grupos dinámicos de los eventos de infracción, como se describe en el reason.

```
{  
  "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 registro NameShadowEvents SizeLimitExcedió la entrada de registro

El tamaño total de los datos de una cosa procesada por la indexación de flotas está limitado a 32 KB. Cuando se infringe este límite para una cosa debido a un evento de sombra con nombre, el NamedShadowEventSizeLimitExceeded se emitirá el tipo de evento.

Ejemplo de entrada de registro NamedShadowEventSizeElimitExceeded

En este ejemplo, se muestra la entrada de registro de unNamedShadowEventSizeLimitExceeded. En este ejemplo, la sombra con nombre (myTestNamedShadow) datos de la cosa (TestThing) se estaba llenando, pero la indexación o filtración dinámica de grupos de NamedShadow se ha omitido, tal como se describe en el reason.

```
{  
  "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 registro NameShadowCount para DynamicGroupQueryLimiTExcedió la entrada de registro

Se procesan un máximo de 5 sombras con nombre por cosa para términos de consulta que no son específicos del origen de datos en grupos dinámicos. Cuando se incumple este límite para una cosa, el NamedShadowCountForDynamicGroupQueryLimitExceeded se emitirá el tipo de evento.

Ejemplo de entrada de registro NamedShadowCountForDynamicGroupQueryLimiteExcedió la entrada de registro

En este ejemplo, se muestra la entrada de registro de `unNamedShadowCountForDynamicGroupQueryLimitExceeded`. En este ejemplo, basado en todos los valores `DynamicGroup` los resultados pueden ser inexactos, tal como se describe en el `reason`.

```
{  
  "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."  
}
```

Común CloudWatch Atributos registros

Todos CloudWatch Las entradas de registro de registros de incluyen estos atributos:

accountId

SusCuenta de AWSID.

eventType

El tipo de evento para el que se generó el registro. El valor del tipo de evento depende del evento que generó la entrada de registro. Cada descripción de entrada de registro incluye el valor de `eventType` para esa entrada de registro.

logLevel

El nivel de registro que se está utilizando. Para obtener más información, consulte [the section called “Niveles de registro” \(p. 432\)](#).

status

El estado de la solicitud.

timestamp

La marca de tiempo de UNIX de cuando el cliente se conectó al agente de mensajes de AWS IoT.

traceId

Un identificador generado aleatoriamente que puede utilizarse para correlacionar todos los registros para una solicitud específica.

Registre las llamadas a la API de AWS IoT con AWS CloudTrail.

AWS IoT está integrado con AWS CloudTrail, un servicio que proporciona un registro de las acciones que realiza un usuario, un rol o un AWS servicio en AWS IoT. CloudTrail captura todas las llamadas a la API de AWS IoT como eventos, incluidas las llamadas de la AWS IoT consola y desde las llamadas de código a la AWS IoT API. Si crea un registro de seguimiento, puede habilitar la entrega continua de eventos de CloudTrail a un bucket de Amazon S3, incluidos los eventos para AWS IoT. Si no configura un registro de

seguimiento, puede ver los eventos más recientes en la CloudTrail Consola enHistorial de eventos. Con la información recopilada por CloudTrail, puede determinar la solicitud que se realizó aAWS IoT, la dirección IP desde la que se realizó la solicitud, quién realizó la solicitud, cuándo se realizó y otros detalles.

Para obtener más información acerca de CloudTrail, consulte la [AWS CloudTrailGuía del usuario de](#).

Información de AWS IoT en CloudTrail

CloudTrail se habilita en su Cuenta de AWS cuando la crea. Cuando se produce actividad enAWS IoT, esa actividad se registra en un CloudTrail evento junto con otrosAWSeventos de servicio enHistorial de eventos. Puede ver, buscar y descargar los últimos eventos de la Cuenta de AWS. Para obtener más información, consulte[Consulta de eventos de con CloudTrail Historial de eventos](#).

Para mantener un registro continuo de eventos en la Cuenta de AWS, incluidos los eventos de AWS IoT, cree un registro de seguimiento. Un rastro permite CloudTrail para enviar archivos de registro a un bucket de Amazon S3. De forma predeterminada, cuando se crea un registro de seguimiento en la consola, el registro de seguimiento se aplica a todas las Región de AWS. El seguimiento registra los eventos de todas las Región de AWS en la partición de AWS y envía los archivos de registro al bucket de Simple Storage Service (Amazon S3) especificado. Puede configurar otrosAWSservicios para analizar y actuar en función de los datos de eventos recopilados en CloudTrail registros. Para obtener más información, consulte:

- [Introducción a la creación de registros de seguimiento](#)
- [Servicios e integraciones compatibles con CloudTrail](#)
- [Configuración de notificaciones de Amazon SNS para CloudTrail](#)
- [Recibir CloudTrail Archivos de registro de de varias regionesyRecepción de archivos de registro de CloudTrail desde varias cuentas](#)

Note

AWS IoTCloudTrail no registra las acciones del plano de datos (lado del dispositivo). Usar CloudWatch para supervisar estas acciones.

En términos generales,AWS IoTLas acciones del plano de control que realizan cambios se registran en CloudTrail. Llamadas tales comoCreateThing,CreateKeysAndCertificate, yUpdateCertificatedejar CloudTrail entradas, mientras que llamadas tales comoListThingsyListTopicRulesNo.

Cada entrada de registro o evento contiene información sobre quién generó la solicitud. La información de identidad del usuario le ayuda a determinar lo siguiente:

- Si la solicitud se realizó con las credenciales raíz o del usuario de IAM.
- Si la solicitud se realizó con credenciales de seguridad temporales de un rol o fue un usuario federado.
- Si la solicitud la realizó otro servicio de AWS.

Para obtener más información, consulte el [Elemento userIdentity de CloudTrail](#).

AWS IoTLas acciones de se documentan en la[AWS IoTReferencia de la API](#).AWS IoT Las acciones inalámbricas se documentan en la[AWS IoTReferencia de la API inalámbrica](#).

Descripción de las entradas de los archivos de registro de AWS IoT

Un registro de seguimiento es una configuración que permite entregar eventos como archivos de registro al bucket de Amazon S3 que especifique. CloudTrail Los archivos de registro de contienen una o varias entradas de registro. Un evento representa una solicitud específica realizada desde un origen y contiene

información sobre la acción solicitada, la fecha y la hora de la acción, los parámetros de la solicitud, etc. CloudTrail Los archivos de registro de no son un rastro de la pila ordenada de las llamadas a la API públicas, por lo que no aparecen en ningún orden específico.

En el siguiente ejemplo, se muestra un CloudTrail entrada de registro que muestra elAttachPolicyaction.

```
{  
    "timestamp": "1460159496",  
    "AdditionalEventData": "",  
    "Annotation": "",  
    "ApiVersion": "",  
    "ErrorCode": "",  
    "ErrorMessage": "",  
    "EventID": "8bfff4fed-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::123456789012: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": ""  
}
```

Reglas para AWS IoT

Las reglas permiten a sus dispositivos interactuar con los servicios de AWS. Las reglas se analizan y las acciones se ejecutan en función del flujo de temas MQTT. Puede utilizar reglas para admitir tareas como las siguientes:

- Incrementar o filtrar los datos recibidos desde un dispositivo.
- Escribir datos recibidos desde un dispositivo en una base de datos de Amazon DynamoDB.
- Guarda un archivo en Amazon S3.
- Enviar una notificación de inserción a todos los usuarios mediante Amazon SNS.
- Publique en una cola de Amazon SQS.
- Invocar una función de Lambda para extraer datos.
- Procesar mensajes de un gran número de dispositivos mediante Amazon Kinesis.
- Enviar datos a Amazon OpenSearch Servicio
- Capturar un CloudWatch Métrica de.
- Cambie un CloudWatch Alarma de.
- Enviar los datos de un mensaje MQTT a Amazon Machine Learning para realizar predicciones basadas en un modelo de Amazon ML.
- Enviar un mensaje a un flujo de entrada de Salesforce IoT
- Enviar datos de mensaje a un canal de AWS IoT Analytics.
- Iniciar la ejecución de una máquina de estado de Step Functions.
- Enviar datos de mensaje a una entrada de AWS IoT Events.
- Envíe datos de mensaje de una propiedad de recurso de AWS IoT SiteWise.
- Envíe datos de mensaje a una aplicación o servicio web.

Sus reglas pueden utilizar mensajes MQTT que pasan por el protocolo de publicación/suscripción compatible con el [the section called “Protocolos de comunicación de dispositivos” \(p. 81\)](#), utilizando el [Basic Ingest \(p. 560\)](#), puede enviar datos del dispositivo de forma segura a la AWS servicios enumerados anteriormente sin incurrir [costos de mensajería](#). (El [Basic Ingest \(p. 560\)](#) optimiza el flujo de datos eliminando el agente de mensajes de publicación/suscripción desde la ruta de adquisición, por lo que resulta más rentable al tiempo que mantiene las características de procesamiento de los datos y de seguridad de AWS IoT.)

Ante AWS IoT puede realizar estas acciones, debe concederle permiso para acceder a su AWS recursos en su nombre. Cuando dichas acciones se ejecuten, se le cobrará la tarifa estándar de los servicios de AWS que utilice.

Contenido

- [Concesión de un AWS IoT regla el acceso que requiere \(p. 473\)](#)
- [Transmisión de los permisos de rol \(p. 474\)](#)
- [Creación de una regla de AWS IoT \(p. 475\)](#)
- [Visualización de las reglas \(p. 479\)](#)
- [Eliminar una regla \(p. 479\)](#)
- [Acciones de reglas de AWS IoT \(p. 479\)](#)
- [Solución de problemas de las reglas \(p. 553\)](#)
- [Acceso a recursos entre cuentas mediante AWS IoT Reglas de \(p. 553\)](#)
- [Control de errores \(acción de error\) \(p. 559\)](#)
- [Reducción de costos de mensajería con Basic Ingest \(p. 560\)](#)

- Referencia de la SQL de AWS IoT (p. 562)

Concesión de unAWS IoT regla el acceso que requiere

Utiliza roles de IAM para controlar elAWSrecursos de a los que cada regla tiene acceso. Antes de crear una regla, primero debe crear un rol de IAM con una política que le permita tener acceso a los requeridosAWSde AWS.IoTasume este rol al ejecutar una regla.

Para crear el rol de IAM yAWS IoTpolítica que otorga unAWS IoTregla el acceso que requiere (AWS CLI)

1. Guarde el siguiente documento de política de confianza, que concedeAWS IoTpermiso para asumir el rol, a un archivo denominado*iot-role-trust.json*.

En este ejemplo se incluye una clave de contexto de condición global para protegerse contra el problema de suplente confuso (p. 343). ParaAWS IoTreglas, suaws:SourceArn debe cumplir el formato: arn:aws:iot:**region:account-id**:*. Asegúrese de que**región**coincide con suAWS IoTRegión y**account-id**coincide con su ID de cuenta de cliente.

```
{  
    "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:*"  
                }  
            }  
        }  
    ]  
}
```

Utilice el comando [create-role](#) para crear un rol de IAM especificando el archivo *iot-role-trust.json*:

```
aws iam create-role --role-name my-iot-role --assume-role-policy-document file://iot-role-trust.json
```

El resultado de este comando tendrá un aspecto similar al siguiente.

```
{  
    "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. Copie el siguiente JSON en un archivo denominado `my-iot-policy.json`.

```
{  
    "Version": "2012-10-17",  
    "Statement": [{  
        "Effect": "Allow",  
        "Action": "dynamodb:*",  
        "Resource": "*"  
    }]  
}
```

Este JSON es un ejemplo de documento de política que concede AWS IoT acceso de administrador a DynamoDB.

Ejecute el comando `create-policy` para conceder a AWS IoT acceso a sus recursos de AWS en el momento en que asuma el rol pasando el archivo `my-iot-policy.json`:

```
aws iam create-policy --policy-name my-iot-policy --policy-document file://my-iot-policy.json
```

Para obtener más información acerca de cómo conceder acceso a los servicios de AWS en las políticas de AWS IoT, consulte [Creación de una regla de AWS IoT \(p. 475\)](#).

La salida del comando `create-policy` contendrá el ARN de la política. Tendrá que adjuntar la política a un rol.

```
{  
    "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. Ejecute el comando `attach-role-policy` para asociar la política al rol:

```
aws iam attach-role-policy --role-name my-iot-role --policy-arn  
"arn:aws:iam::123456789012:policy/my-iot-policy"
```

Transmisión de los permisos de rol

La definición de una regla es un rol de IAM que concede permiso para tener acceso a los recursos especificados en la acción de la regla. El motor de reglas asume dicho rol cuando se activa la acción de la regla. El rol tiene que estar definido en el mismo Cuenta de AWS como regla general.

De hecho, cuando crea o sustituye una regla, está transfiriendo un rol al motor de reglas. El usuario que realiza esta operación necesita el permiso `iam:PassRole`. Para asegurarse de que dispone de dicho permiso, cree una política que conceda el `iam:PassRole` adjunte a su usuario de IAM. La política siguiente muestra cómo conceder un permiso `iam:PassRole` para un rol.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "Stmt1",  
            "Effect": "Allow",  
            "Action": [  
                "iam:PassRole"  
            ],  
            "Resource": [  
                "arn:aws:iam::123456789012:role/myRole"  
            ]  
        }  
    ]  
}
```

En este ejemplo de política, se concede el permiso `iam:PassRole` para el rol `myRole`. El rol se especifica mediante el ARN del rol. Debe asociar esta política al usuario o al rol de IAM al que pertenezca el usuario. Para obtener más información, consulte [Uso de políticas administradas](#).

Note

Las funciones de Lambda utilizan una política basada en recursos. Esta política está directamente asociada a la función de Lambda en sí. Cuando crea una regla que invoca una función de Lambda, no transmite un rol, por lo que el usuario que crea la regla no necesita la `iam:PassRole` permiso. Para obtener más información acerca de la autorización de las funciones de Lambda, consulte [Conceder permisos mediante una política de recursos](#).

Creación de una regla de AWS IoT

Puede configurar las reglas para dirigir datos desde los objetos conectados. Las reglas constan de los elementos siguientes:

Nombre de la regla

El nombre de la regla.

Note

No es recomendable utilizar información de identificación personal en los nombres de regla.

Descripción opcional

Descripción textual de la regla.

Note

No es recomendable utilizar información de identificación personal en las descripciones de regla.

Instrucción SQL

Sintaxis de SQL simplificada para filtrar los mensajes recibidos sobre un tema MQTT e insertar los datos en otro punto. Para obtener más información, consulte [Referencia de la SQL de AWS IoT \(p. 562\)](#).

Versión de SQL

Versión del motor de reglas SQL que debe utilizarse al evaluar la regla. Aunque esta propiedad es opcional, recomendamos encarecidamente que especifique la versión de SQL. La AWS IoT CoreLa consola establece esta propiedad como 2016-03-23 de forma predeterminada. Si no se define esta propiedad, como en un AWS CLI comando o un AWS CloudFormation plantilla, 2015-10-08 se utiliza. Para obtener más información, consulte [Versiones de SQL \(p. 625\)](#).

Una o varias acciones

Las acciones que realiza AWS IoT al ejecutar la regla. Por ejemplo, puede insertar datos en una tabla de DynamoDB, escribir datos en un bucket de Amazon S3, publicar en un tema de Amazon SNS o invocar una función de Lambda.

Una acción de error

La acción AWS IoT se realiza cuando no es posible realizar una acción de la regla.

Cuando cree una regla, debe tener en cuenta la cantidad de datos que publica en los temas. Si crea reglas que contienen un patrón de tema con comodín, es posible que coincidan con una gran cantidad de mensajes y que necesite aumentar la capacidad de los recursos de AWS se que utilizan en las acciones de destino. Asimismo, si crea una regla para volver a publicar que contenga un patrón de tema con comodín, puede acabar teniendo una regla circular que genere un bucle infinito.

Note

La creación y la actualización de reglas son acciones de nivel de administrador. Todo usuario que tenga permiso para crear o actualizar reglas podrá tener acceso a los datos procesados por las reglas.

Para crear una regla (AWS CLI)

Ejecute el comando [create-topic-rule](#) para crear una regla:

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

Ejemplo de archivo de carga con una regla que inserta todos los mensajes enviados a laiot/testtema en la tabla de DynamoDB especificada. La instrucción SQL filtra los mensajes, y el ARN del rol concedeAWS IoTpermiso para escribir en la tabla de 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()}"
            }
        }
    ]
}
```

A continuación, se muestra un ejemplo de archivo de carga con una regla que inserta todos los mensajes enviados al tema iot/test en el bucket de S3 especificado. La instrucción SQL filtra los mensajes, y el ARN del rol concede a permiso paraAWS IoTpermiso para escribir en el bucket de 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": "topic"
            }
        }
    ]
}
```

```
        "key": "myS3Key"
    }
}
]
```

Ejemplo de archivo de carga con una regla que inserta datos en Amazon OpenSearch Servicio de :

```
{
    "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()}"
            }
        }
    ]
}
```

Ejemplo de archivo de carga con una regla que invoca una función Lambda:

```
{
    "sql": "expression",
    "ruleDisabled": false,
    "awsIotSqlVersion": "2016-03-23",
    "actions": [
        {
            "lambda": {
                "functionArn": "arn:aws:lambda:us-west-2:123456789012:function:my-lambda-function"
            }
        }
    ]
}
```

Ejemplo de archivo de carga con una regla que publica en un tema de 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"
            }
        }
    ]
}
```

Ejemplo de archivo de carga con una regla que vuelve a publicar en otro tema MQTT:

```
{
    "sql": "expression",
    "ruleDisabled": false,
    "awsIotSqlVersion": "2016-03-23",
    "actions": [
        {
            "republish": {

```

```
        "topic": "my-mqtt-topic",
        "roleArn": "arn:aws:iam::123456789012:role/my-iot-role"
    }
}
}
```

Ejemplo de archivo de carga con una regla que inserta datos en un flujo de 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"
      }
    }
}
```

Ejemplo de archivo de carga con una regla que utiliza Amazon Machine Learning para volver a publicar en un tema si los datos de la carga de MQTT se clasifican 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"
      }
    }
}
```

A continuación se incluye un archivo de carga de ejemplo con una regla que publica mensajes en un flujo de entrada de 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"
      }
    }
}
```

Ejemplo de archivo de carga con una regla que inicia la ejecución de una máquina de estado de 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"  
        }  
    }]  
}
```

Visualización de las reglas

Ejecute el comando [list-topic-rules](#) para visualizar una lista de sus reglas:

```
aws iot list-topic-rules
```

Ejecute el comando [get-topic-rule](#) para obtener información acerca de una regla:

```
aws iot get-topic-rule --rule-name myrule
```

Eliminar una regla

Cuando ya no necesite una regla, puede eliminarla.

Para eliminar una regla (AWS CLI)

Ejecute el comando [delete-topic-rule](#) para eliminar una regla:

```
aws iot delete-topic-rule --rule-name myrule
```

Acciones de reglas de AWS IoT

AWS IoTLas acciones de las reglas especifican qué hacer cuando se activa una regla. Puede definir acciones para enviar datos a una base de datos de Amazon DynamoDB, enviar datos a Amazon Kinesis Data Streams, invocar unAWS Lambdafunción, y así sucesivamente.AWS IoTadmite las siguientes acciones enRegión de AWSs donde el servicio de la acción está disponible.

Acción de la regla	Descripción	Nombre de la API
Apache Kafka (p. 481)	Envía un mensaje a un clúster de Apache Kafka.	kafka
Alarms de CloudWatch (p. 489)	Cambia el estado de un Amazon CloudWatch Alarma de.	cloudwatchAlarm
CloudWatch Logs (p. 490)	Envía un mensaje a Amazon CloudWatch Registros.	cloudwatchLogs
Métricas de CloudWatch (p. 491)	Envía un mensaje a un CloudWatch Métrica de.	cloudwatchMetric
DynamoDB (p. 493)	Envía un mensaje a una tabla de DynamoDB.	dynamoDB

Acción de la regla	Descripción	Nombre de la API
DynamoDBv2 (p. 495)	Envía datos de mensajes a varias columnas de una tabla de DynamoDB.	dynamoDBv2
Elasticsearch (p. 497)	Envía un mensaje a un punto de enlace de Elasticsearch.	elasticsearch
HTTP (p. 498)	Publica un mensaje en un punto de enlace HTTPS.	http
IoT Analytics (p. 525)	Envía un mensaje a unAWS IoT Analyticscanal de.	iotAnalytics
IoT Events (p. 527)	Envía un mensaje a unAWS IoT Eventsentrada.	iotEvents
IoT SiteWise (p. 528)	Envía datos de mensajes aAWS IoT SiteWisepropiedades de activos.	iotSiteWise
Kinesis Data Firehose (p. 532)	Envía un mensaje a un flujo de entrega de Kinesis Data Firehose.	firehose
Kinesis Data Streams (p. 534)	Envía un mensaje a un flujo de datos de Kinesis.	kinesis
Lambda (p. 536)	Invoca una función Lambda con datos de mensaje como entrada.	lambda
OpenSearch (p. 538)	Envía un mensaje a Amazon OpenSearch Punto de enlace de servicio.	OpenSearch
Republish (p. 540)	Vuelve a publicar un mensaje en otro tema MQTT.	republish
S3 (p. 541)	Almacena un mensaje en un bucket de Amazon Simple Storage Service (Amazon S3)	s3
Salesforce IoT IoT (p. 543)	Envía un mensaje a un flujo de entrada de Salesforce IoT.	salesforce
SNS (p. 543)	Publica un mensaje como notificación push de Amazon Simple Notification Service (Amazon SNS)	sns
SQS (p. 545)	Envía un mensaje a una cola de Amazon Simple Queue Service (Amazon SQS)	sqrs
Step Functions (p. 547)	Inicia unAWS Step Functionsmáquina de estado de.	stepFunctions

Acción de la regla	Descripción	Nombre de la API
the section called "Timestream" (p. 548)	Envía un mensaje a una tabla de base de datos de Amazon Timestream.	timestream

Notas

- Debe definir la regla en la misma Región de AWS como recurso de otro servicio, de modo que la acción de regla pueda interactuar con ese recurso.
- Es posible que el motor de reglas de AWS IoT realice varios intentos para ejecutar una acción en caso de errores intermitentes. Si todos los intentos fallan, el mensaje se descarta y el error está disponible en el CloudWatch registros. Es posible especificar una acción de error para cada regla que se invoca cuando se produce un error. Para obtener más información, consulte [Control de errores \(acción de error\) \(p. 559\)](#).
- Algunas acciones de regla desencadenan acciones en servicios que se integran con AWS Key Management Service (AWS KMS) para admitir el cifrado de datos en reposo. Si utilizas un cliente administrado AWS KMS key (clave KMS) para cifrar los datos en reposo, el servicio debe tener permiso para poder usar la clave KMS en nombre del autor de la llamada. Consulte los temas de cifrado de datos en la guía del servicio apropiado para obtener información acerca de cómo administrar los permisos de la clave de KMS administrada por el cliente. Para obtener más información acerca de las claves KMS administradas por el cliente, consulte [AWS Key Management Service Conceptos de AWS Key Management Service Guía para desarrolladores](#).

Apache Kafka

La acción Apache Kafka (Kafka) envía mensajes directamente a los clústeres de Apache Kafka de Managed Streaming for Apache Kafka (Amazon MSK) o clústeres de Apache Kafka autoadministrados para análisis de datos y análisis de datos.

Note

En este tema se supone que se conocen la plataforma Apache Kafka y los conceptos relacionados. Para obtener más información acerca de Apache Kafka, consulte [Apache Kafka](#).

Requisitos

Esta acción de regla debe cumplir estos requisitos:

- Un rol de IAM que AWS IoT puede suponer que se lleva a cabo `ec2:CreateNetworkInterface,ec2:DescribeNetworkInterfaces,ec2:CreateNetworkInterfacePermission,ec2:DescribeSecurityGroups` operaciones. Este rol crea y administra interfaces de red elásticas en Amazon Virtual Private Cloud para comunicarse con su agente de Kafka. Para obtener más información, consulte [Concesión de un AWS IoT regla el acceso que requiere \(p. 473\)](#).

En el navegador AWS IoT consola, puede elegir o crear un rol para permitir AWS IoT Core para llevar a cabo esta acción de regla de.

Para obtener más información acerca de las interfaces de red, consulte [Interfaces de red elásticas](#) en la Guía del usuario de Amazon EC2.

La política asociada al rol que especifique debe tener el aspecto que se muestra en el siguiente ejemplo.

```
{
```

```
"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": "*"
}
]
```

- Si usa AWS Secrets Manager para almacenar las credenciales necesarias para conectarse a su agente de Kafka, debe crear un rol de IAM que AWS IoT Core puede suponer que se lleva a cabo en `secretsmanager:GetSecretValue` y `secretsmanager:DescribeSecret` operaciones.

La política asociada al rol que especifique debe tener el aspecto que se muestra en el siguiente ejemplo.

```
{
    "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-*"
            ]
        }
    ]
}
```

- Puede ejecutar sus clústeres de Apache Kafka dentro de Amazon Virtual Private Cloud. Debe crear un destino de nube virtual privada (VPC) y utilizar una puerta de enlace NAT en sus subredes para reenviar mensajes desde AWS IoT a un clúster público de Kafka. La AWS IoT Rules engine crea una interfaz de red en cada una de las subredes enumeradas en el destino de la VPC para enrutar el tráfico directamente a la VPC. Al crear un destino de la VPC, la AWS IoT Rules engine crea automáticamente una acción de regla de VPC. Para obtener más información acerca de las acciones de reglas de la VPC, consulte [Destinos de nube virtual privada \(VPC\) \(p. 486\)](#).
- Si utilizas un cliente administrado AWS KMS key (clave KMS) para cifrar los datos en reposo, el servicio debe tener permiso para poder usar la clave KMS en nombre del autor de la llamada. Para obtener más información, consulta [Cifrado de Amazon MSK](#) en la Guía para desarrolladores de Amazon Managed Streaming for Apache Kafka.

Parámetros

Al crear un AWS IoT con esta acción, debe especificar la información siguiente:

DestinationArn

El nombre de recurso de Amazon (ARN) del destino de la VPC. Para obtener información acerca de cómo crear un destino de VPC, consulte [Destinos de nube virtual privada \(VPC\) \(VPC\) \(p. 486\)](#).

tema

El tema de Kafka para enviar mensajes al corredor de Kafka.

Puede sustituir este campo mediante una plantilla de sustitución. Para obtener más información, consulte [the section called “Plantillas de sustitución” \(p. 622\)](#).

clave (opcional)

La clave del mensaje de Kafka.

Puede sustituir este campo mediante una plantilla de sustitución. Para obtener más información, consulte [the section called “Plantillas de sustitución” \(p. 622\)](#).

partición (opcional)

La partición de mensajes Kafka.

Puede sustituir este campo mediante una plantilla de sustitución. Para obtener más información, consulte [the section called “Plantillas de sustitución” \(p. 622\)](#).

ClientProperties

Objeto que define las propiedades del cliente productor Apache Kafka.

Tacks (opcional)

El número de confirmaciones que el productor requiere que el servidor haya recibido antes de considerar que se ha completado una solicitud.

Si especifica 0 como valor, el productor no esperará ningún acuse de recibo del servidor. Si el servidor no recibe el mensaje, el productor no volverá a intentar enviarlo.

Valores válidos: 0, 1 El valor predeterminado es 1.

bootstrap.servers

Lista de pares de host y puertos (host1:port1,host2:port2, etc.) que se utiliza para establecer la conexión inicial con el clúster de Kafka.

compression.type (opcional)

Tipo de compresión de todos los datos generados por el productor.

Valores válidos: none,gzip,snappy,lz4,zstd. El valor predeterminado esnone.

security.protocol

Protocolo de seguridad utilizado para adjuntar a su agente Kafka.

Valores válidos: SSL,SASL_SSL. El valor predeterminado esSSL.

key.serializer

Especifica cómo girar los objetos clave que proporciona con elProducerRecorden bytes.

Valor válido:StringSerializer.

value.serializer

Especifica cómo convertir los objetos de valor que proporciona con elProducerRecorden bytes.

Valor válido:ByteBufferSerializer.

ssl.truststore

El archivo truststore en formato base64 o la ubicación del archivo truststore en [AWS Secrets Manager](#). Este valor no es obligatorio si las autoridades de certificados (CA) de Amazon confían en su almacén de confianza.

Este campo admite plantillas de sustitución. Si utiliza Secrets Manager para almacenar las credenciales necesarias para conectarse a su agente de Kafka, puede utilizar la función SQL `get_secret` para recuperar el valor de este campo. Para obtener más información sobre las plantillas de sustitución, consulte [the section called “Plantillas de sustitución” \(p. 622\)](#). Para obtener más información acerca de la función SQL `get_secret`, consulte [the section called “get_secret \(secretId, tipo secreto, clave, roleArn\)” \(p. 591\)](#). Si el almacén de confianza tiene la forma de un archivo, utilice la `SecretBinary` parámetro de. Si el almacén de confianza tiene la forma de una cadena, utilice el `SecretString` parámetro de.

El tamaño máximo de este valor es 65 KB.

ssl.truststore.password

La contraseña del almacén de confianza. Este valor solo es obligatorio si ha creado una contraseña para el almacén de confianza.

ssl.keystore

El archivo de almacén de claves. Este valor es obligatorio cuando se especifica `SSL` como valor `desecurity.protocol`.

Este campo admite plantillas de sustitución. Debe utilizar Secrets Manager para almacenar las credenciales necesarias para conectarse a su agente de Kafka. Utilice la función SQL `get_secret` para recuperar el valor de este campo. Para obtener más información sobre las plantillas de sustitución, consulte [the section called “Plantillas de sustitución” \(p. 622\)](#). Para obtener más información acerca de la función SQL `get_secret`, consulte [the section called “get_secret \(secretId, tipo secreto, clave, roleArn\)” \(p. 591\)](#). Utilice el parámetro `SecretBinary`.

ssl.keystore.password

La contraseña del almacén de claves. Este valor es obligatorio si especifica un valor para `ssl.keystore`.

El valor de este campo puede ser texto sin formato. Este campo también admite plantillas de sustitución. Debe utilizar Secrets Manager para almacenar las credenciales necesarias para conectarse a su agente de Kafka. Utilice la función SQL `get_secret` para recuperar el valor de este campo. Para obtener más información sobre las plantillas de sustitución, consulte [the section called “Plantillas de sustitución” \(p. 622\)](#). Para obtener más información acerca de la función SQL `get_secret`, consulte [the section called “get_secret \(secretId, tipo secreto, clave, roleArn\)” \(p. 591\)](#). Utilice el parámetro `SecretString`.

ssl.key.password

La contraseña de la clave privada del archivo del almacén de claves.

Este campo admite plantillas de sustitución. Debe utilizar Secrets Manager para almacenar las credenciales necesarias para conectarse a su agente de Kafka. Utilice la función SQL `get_secret` para recuperar el valor de este campo. Para obtener más información sobre las plantillas de sustitución, consulte [the section called “Plantillas de sustitución” \(p. 622\)](#). Para obtener más información acerca de la función SQL `get_secret`, consulte [the section called “get_secret \(secretId, tipo secreto, clave, roleArn\)” \(p. 591\)](#). Utilice el parámetro `SecretString`.

sasl.mecanismo

El mecanismo de seguridad utilizado para conectarse a su agente Kafka. Este valor es obligatorio cuando se especifica `SASL_SSL` para `security.protocol`.

Valores válidos: PLAIN, SCRAM-SHA-512, GSSAPI.

Note

SCRAM-SHA-512 es el único mecanismo de seguridad compatible en las regiones cn-north-1, cn-northwest-1, us-gov-east-1 y us-gov-west-1.

sasl.plain.nombre de usuario

El nombre de usuario utilizado para recuperar la cadena secreta de Secrets Manager. Este valor es obligatorio cuando se especifica `SASL_SSLparasecurity.protocol=PLAINparasasl.mechanism`.

sasl.plain.password

La contraseña utilizada para recuperar la cadena secreta de Secrets Manager. Este valor es obligatorio cuando se especifica `SASL_SSLparasecurity.protocol=PLAINparasasl.mechanism`.

sasl.scram.username

El nombre de usuario utilizado para recuperar la cadena secreta de Secrets Manager. Este valor es obligatorio cuando se especifica `SASL_SSLparasecurity.protocol=SCRAM-SHA-512parasasl.mechanism`.

sasl.scram.password

La contraseña utilizada para recuperar la cadena secreta de Secrets Manager. Este valor es obligatorio cuando se especifica `SASL_SSLparasecurity.protocol=SCRAM-SHA-512parasasl.mechanism`.

sasl.kerberos.keytab

El archivo keytab para la autenticación Kerberos en Secrets Manager. Este valor es obligatorio cuando se especifica `SASL_SSLparasecurity.protocol=GSSAPIparasasl.mechanism`.

Este campo admite plantillas de sustitución. Debe utilizar Secrets Manager para almacenar las credenciales necesarias para conectarse a su agente de Kafka. Utilice la función SQL `get_secret` para recuperar el valor de este campo. Para obtener más información sobre las plantillas de sustitución, consulte [the section called “Plantillas de sustitución” \(p. 622\)](#). Para obtener más información acerca de la función SQL `get_secret`, consulte [the section called “get_secret \(secretId, tipo secreto, clave, roleArn\)” \(p. 591\)](#). Utilice el parámetro `SecretBinary`.

sasl.kerberos.service.name

El nombre principal de Kerberos bajo el que se ejecuta Apache Kafka. Este valor es obligatorio cuando se especifica `SASL_SSLparasecurity.protocol=GSSAPIparasasl.mechanism`.

sasl.kerberos.krb5.kdc

Nombre de host del centro de distribución de claves (KDC) al que se conecta el cliente productor Apache Kafka. Este valor es obligatorio cuando se especifica `SASL_SSLparasecurity.protocol=GSSAPIparasasl.mechanism`.

sasl.kerberos.krb5.realm

El ámbito al que se conecta su cliente productor Apache Kafka. Este valor es obligatorio cuando se especifica `SASL_SSLparasecurity.protocol=GSSAPIparasasl.mechanism`.

sasl.kerberos.principal

La identidad exclusiva de Kerberos a la que Kerberos puede asignar tickets para acceder a servicios compatibles con Kerberos. Este valor es obligatorio cuando se especifica `SASL_SSLparasecurity.protocol=GSSAPIparasasl.mechanism`.

Ejemplos

En el siguiente ejemplo de JSON se define una acción Apache Kafka en unAWS IoTregla.

```
{  
    "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 la configuración de Kerberos

- El centro de distribución de claves (KDC) debe resolverse a través del Sistema de nombres de dominio privado (DNS) dentro de la VPC de destino. Un posible enfoque es agregar la entrada DNS de KDC a una zona alojada privada. Para obtener más información acerca de este enfoque, consulte [Uso de zonas alojadas privadas](#).
- Cada VPC debe tener habilitada la resolución DNS. Para obtener más información, consulte [Utilización de DNS con su VPC](#).
- Los grupos de seguridad de la interfaz de red y los grupos de seguridad a nivel de instancia en el destino de la VPC deben permitir el tráfico desde la VPC en los siguientes puertos.
 - Tráfico TCP en el puerto de escucha del agente de arranque (a menudo 9092, pero debe estar dentro del rango 9000 - 9100)
 - Tráfico TCP y UDP en el puerto 88 para el KDC
- SCRAM-SHA-512es el único mecanismo de seguridad compatible en las regiones cn-north-1, cn-northwest-1, us-gov-east-1 y us-gov-west-1.

Destinos de nube virtual privada (VPC) (VPC)

La acción de la regla de Apache Kafka enruta los datos a un clúster de Apache Kafka en una instancia de Amazon Virtual Private Cloud (Amazon VPC). La configuración de VPC utilizada por la acción de regla Apache Kafka se activa automáticamente cuando especifica el destino de la VPC para la acción de regla.

Un destino de VPC contiene una lista de subredes dentro de la VPC. El motor de reglas crea una elastic network interface en cada subred especificada en esta lista. Para obtener más información acerca de las interfaces de red, consulte [Interfaces de red elásticas](#) en la Guía del usuario de Amazon EC2.

Requisitos y consideraciones

- Si utilizas un clúster de Apache Kafka autoadministrado al que se accederá mediante un endpoint público a través de Internet:
 - Debe crear una puerta de enlace NAT para las instancias de las subredes. La gateway NAT tiene una dirección IP pública que puede conectarse a Internet, lo que permite al motor de reglas reenviar mensajes al clúster público de Kafka.
 - Como alternativa menos costosa a las puertas de enlace NAT, puede asignar una dirección IP elástica con las interfaces de red elásticas (ENI) creadas por el destino de la VPC. Los grupos de seguridad que utiliza deben configurarse para bloquear el tráfico entrante.

Note

Si el destino de la VPC está deshabilitado y, a continuación, vuelve a habilitar, debe volver a asociar las IP elásticas a los nuevos ENI.

- Si un destino de regla de tema de VPC no recibe tráfico durante 30 días seguidos, se deshabilitará.
- Si cambia algún recurso utilizado por el destino de la VPC, el destino se deshabilitará y no se podrá utilizar.
- Algunos cambios que pueden deshabilitar un destino de VPC incluyen: eliminar la VPC, subredes, grupos de seguridad o el rol utilizado; modificar el rol para que deje de tener los permisos necesarios; y deshabilitar el destino.

Precios

A efectos de precios, se mide una acción de regla de VPC además de la acción que envía un mensaje a un recurso cuando el recurso se encuentra en la VPC. Para obtener información sobre los precios, consulte [Precios de AWS IoT Core](#).

Creación de destinos de reglas de tema de la nube virtual privada (VPC)

Para crear un destino de nube virtual privada (VPC) mediante el [CreateTopicRuleDestination](#) API o la [AWS IoT Core](#) consola de .

Cuando cree un destino de VPC, debe especificar la información siguiente.

`vpclId`

El ID exclusivo del destino de la VPC.

`subnetIds`

Lista de subredes en las que el motor de reglas crea interfaces de red elásticas. El motor de reglas asigna una única interfaz de red para cada subred de la lista.

`Groups` de seguridad (opcional)

Lista de grupos de seguridad que se van a aplicar a las interfaces de red.

`roleArn`

El nombre de recurso de Amazon (ARN) de un rol que tiene permiso para crear interfaces de red en su nombre.

Este ARN debe tener adjunta una política similar al siguiente ejemplo.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "ec2:CreateNetworkInterface",  
                "ec2:DescribeNetworkInterfaces",  
                "ec2:CreateNetworkInterfacePermission",  
                "ec2:DeleteNetworkInterface",  
                "ec2:DescribeSecurityGroups",  
                "ec2:DescribeSubnets",  
                "ec2:DescribeVpcs",  
                "ec2:DescribeVpcAttribute"  
            ],  
            "Resource": "*"  
        }  
    ]  
}
```

Creación de un destino de VPC medianteAWS CLI

En el ejemplo siguiente se muestra cómo crear un destino de VPC medianteAWS 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"}'
```

Cuando ejecuta este comando, el estado del destino de la VPC seráIN_PROGRESS. Transcurridos unos minutos, su estado cambiará a:ERROR(si el comando no tiene éxito) oENABLED. Cuando el estado de destino esENABLED, está lista para su uso.

Puede utilizar el siguiente comando para obtener el estado del destino de su VPC.

```
aws --region region iot get-topic-rule-destination --arn "VPCDestinationARN"
```

Creación de un destino de la VPC mediante laAWS IoT Coreconsola

Los siguientes pasos describen cómo crear un destino de VPC mediante laAWS IoT Coreconsola de .

1. Vaya a la consola de AWS IoT Core. En el panel de la izquierda, en elActopestaña, elijaDestinos.
2. Escriba valores para los siguientes campos.
 - ID de VPC
 - ID de subred
 - Security Group
3. Seleccione un rol que tenga los permisos necesarios para crear interfaces de red. La directiva de ejemplo anterior contiene estos permisos.

Cuando el estado de destino de la VPC esENABLED, está lista para su uso.

Alarmas de CloudWatch

La CloudWatch Alarma de `cloudwatchAlarm` cambia el estado de un Amazon CloudWatch Alarma de. Puede especificar el motivo del cambio de estado y el valor de esta llamada.

Requisitos

Esta acción de regla tiene los siguientes requisitos:

- Un rol de IAM que AWS IoT puede suponer que se lleva a cabo el `cloudwatch:SetAlarmState`. Para obtener más información, consulte [Concesión de unAWS IoT regla el acceso que requiere \(p. 473\)](#).

En el navegador AWS IoT consola, puede elegir o crear un rol para permitir AWS IoT para llevar a cabo esta acción de regla de.

Parámetros

Al crear unAWS IoT con esta acción, debe especificar la información siguiente:

`alarmName`

La CloudWatch nombre de alarma.

Admite [Plantillas de sustitución \(p. 622\)](#): API y AWS CLIs sólo

`stateReason`

El motivo del cambio de alarma.

Admite [Plantillas de sustitución \(p. 622\)](#): Sí

`stateValue`

El valor del estado de alarma. Valores válidos: OK, ALARM, INSUFFICIENT_DATA.

Admite [Plantillas de sustitución \(p. 622\)](#): Sí

`roleArn`

El rol de IAM que permite tener acceso a la CloudWatch Alarma de. Para obtener más información, consulte [Requisitos \(p. 489\)](#).

Admite [Plantillas de sustitución \(p. 622\)](#): No

Ejemplos

En el siguiente ejemplo de JSON se define un CloudWatch acción de alarma en unAWS IoT regla.

```
{  
    "topicRulePayload": {  
        "sql": "SELECT * FROM 'some/topic'",  
        "ruleDisabled": false,  
        "awsIotSqlVersion": "2016-03-23",  
        "actions": [  
            {  
                "cloudwatchAlarm": {  
                    "alarmName": "IoTAlarm",  
                    "stateValue": "ALARM",  
                    "stateReason": "CloudWatch Alarma de."  
                }  
            }  
        ]  
    }  
}
```

```
        "stateReason": "Temperature stabilized.",
        "stateValue": "OK",
        "roleArn": "arn:aws:iam::123456789012:role/aws_iot_cw"
    }
}
}
```

Véase también

- [¿Qué es Amazon CloudWatch?](#)en laAmazon CloudWatch Guía del usuario de
- [Uso de alarmas de Amazon CloudWatch](#)en laAmazon CloudWatch Guía del usuario de

CloudWatch Logs

La CloudWatch Registros (`cloudwatchLogs`) envía datos a Amazon CloudWatch Registros. Puede especificar el grupo de registros de al que la acción va a enviar los datos.

Requisitos

Esta acción de regla debe cumplir estos requisitos:

- Un rol de IAM queAWS IoT puede suponer que se lleva a cabo `elogs:CreateLogStream`, `logs:DescribeLogStreams`, y `logs:PutLogEvents` operaciones. Para obtener más información, consulte [Concesión de unAWS IoT regla el acceso que requiere \(p. 473\)](#).

En el navegadorAWS IoT consola, puede elegir o crear un rol para permitirAWS IoT para llevar a cabo esta acción de regla de.

- Si utilizas un cliente administradoAWS KMS key(clave KMS) para cifrar los datos de registro en CloudWatch Registros, el servicio debe tener permiso para poder usar la clave KMS en nombre del autor de la llamada. Para obtener más información, consulte [Cifrado de datos de registro en CloudWatch Registros conAWS KMS](#)en laAmazon CloudWatch Guía del usuario de Logs.

Parámetros

Al crear unAWS IoT con esta acción, debe especificar la información siguiente:

`logGroupName`

La CloudWatch grupo de registros de al que la acción envía los datos.

Admite[Plantillas de sustitución \(p. 622\)](#): API yAWS CLIsólo

`roleArn`

El rol de IAM que permite tener acceso a la CloudWatch grupo de registros de. Para obtener más información, consulte [Requisitos \(p. 490\)](#).

Admite[Plantillas de sustitución \(p. 622\)](#): No

Ejemplos

En el siguiente ejemplo de JSON se define un CloudWatch Registra la acción en unAWS IoT regla.

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

Véase también

- [¿Qué es Amazon? CloudWatch ¿Registros?](#)en laAmazon CloudWatch Guía del usuario de Logs

Métricas de CloudWatch

La CloudWatch Métrica de (`cloudwatchMetric`) captura una métrica de Amazon CloudWatch. Puede especificar el espacio de nombres, el nombre, el valor, la unidad y la marca de tiempo de la métrica.

Requisitos

Esta acción de regla debe cumplir estos requisitos:

- Un rol de IAM queAWS IoT puede suponer que se lleva a cabo el`cloudwatch:PutMetricData`. Para obtener más información, consulte [Concesión de unAWS IoT regla el acceso que requiere \(p. 473\)](#).

En el navegadorAWS IoT consola, puede elegir o crear un rol para permitirAWS IoT para llevar a cabo esta acción de regla de.

Parámetros

Al crear unAWS IoT con esta acción, debe especificar la información siguiente:

`metricName`

La CloudWatch nombre de métrica.

Admite[Plantillas de sustitución \(p. 622\)](#): Sí

`metricNamespace`

La CloudWatch El nombre del espacio de nombres de la métrica de .

Admite[Plantillas de sustitución \(p. 622\)](#): Sí

`metricUnit`

La unidad métrica compatible con CloudWatch.

Admite[Plantillas de sustitución \(p. 622\)](#): Sí

metricValue

Una cadena que contiene el CloudWatch valor de métrica.

Admite[Plantillas de sustitución \(p. 622\)](#): Sí

metricTimestamp

(Opcional) Una cadena que contiene la marca de tiempo Unix. El valor predeterminado es la época actual de Unix.

Admite[Plantillas de sustitución \(p. 622\)](#): Sí

roleArn

El rol de IAM que permite tener acceso a la CloudWatch Métrica de. Para obtener más información, consulte [Requisitos \(p. 491\)](#).

Admite[Plantillas de sustitución \(p. 622\)](#): No

Ejemplos

En el siguiente ejemplo de JSON se define un CloudWatch Acción de métrica deAWS IoTregla.

```
{  
    "topicRulePayload": {  
        "sql": "SELECT * FROM 'some/topic'",  
        "ruleDisabled": false,  
        "awsIotSqlVersion": "2016-03-23",  
        "actions": [  
            {  
                "cloudwatchMetric": {  
                    "metricName": "IoMetric",  
                    "metricNamespace": "IoNamespace",  
                    "metricUnit": "Count",  
                    "metricValue": "1",  
                    "metricTimestamp": "1456821314",  
                    "roleArn": "arn:aws:iam::123456789012:role/aws_iot_cw"  
                }  
            }  
        ]  
    }  
}
```

En el siguiente ejemplo de JSON se define un CloudWatch acción métrica con plantillas de sustitución en unAWS IoTregla.

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

```
        }
    ]
}
```

Véase también

- [¿Qué es Amazon CloudWatch?](#)en laAmazon CloudWatch Guía del usuario de
- [Uso de métricas de Amazon CloudWatch](#)en laAmazon CloudWatch Guía del usuario de

DynamoDB

El DynamoDB (`dynamodb`) escribe todo un mensaje MQTT o parte de él en una tabla de Amazon DynamoDB.

Puede seguir un tutorial que le muestra cómo crear y probar una regla con una acción de DynamoDB. Para obtener más información, consulte [Tutorial: Almacenamiento de datos de dispositivos en una tabla de DynamoDB \(p. 209\)](#).

Note

Esta regla escribe datos no JSON en DynamoDB como datos binarios. La consola de DynamoDB mostrará los datos como texto codificado en Base64.

Requisitos

Esta acción de regla debe cumplir estos requisitos:

- Un rol de IAM queAWS IoT puede suponer que se lleva a cabo el `dynamodb:PutItem`. Para obtener más información, consulte [Concesión de unAWS IoT regla el acceso que requiere \(p. 473\)](#).

En el navegadorAWS IoT consola, puede elegir o crear un rol para permitirAWS IoT para llevar a cabo esta acción de regla de.

- Si utilizas un cliente administradoAWS KMS key(clave KMS) para cifrar los datos en reposo de en DynamoDB, el servicio debe tener permiso para poder usar la clave KMS en nombre del autor de la llamada. Para obtener más información, consulte [Clave KMS administrada por el cliente](#)en laGuía de introducción a Amazon DynamoDB.

Parámetros

Al crear unAWS IoT con esta acción, debe especificar la información siguiente:

`tableName`

El nombre de la tabla de DynamoDB.

Admite [Plantillas de sustitución \(p. 622\)](#): API yAWS CLIsólo

`hashKeyField`

El nombre de la clave hash (también denominada clave de partición).

Admite [Plantillas de sustitución \(p. 622\)](#): API yAWS CLIsólo

hashKeyType

(Opcional) El tipo de datos de la clave hash (también denominada clave de partición). Valores válidos: STRING, NUMBER.

Admite[Plantillas de sustitución \(p. 622\)](#): API y AWS CLIsólo

hashKeyValue

El valor de la clave hash. Considere la posibilidad de utilizar una plantilla de sustitución, como \${topic()}o\${timestamp()}.

Admite[Plantillas de sustitución \(p. 622\)](#): Sí

rangeKeyField

(Opcional) nombre de la clave de rango (también denominada clave de ordenación).

Admite[Plantillas de sustitución \(p. 622\)](#): API y AWS CLIsólo

rangeKeyType

(Opcional) El tipo de datos de la clave de rango (también denominada clave de ordenación). Valores válidos: STRING, NUMBER.

Admite[Plantillas de sustitución \(p. 622\)](#): API y AWS CLIsólo

rangeKeyValue

(Opcional) El valor de la clave de rango. Considere la posibilidad de utilizar una plantilla de sustitución, como \${topic()}o\${timestamp()}.

Admite[Plantillas de sustitución \(p. 622\)](#): Sí

payloadField

(Opcional) nombre de la columna en la que se escribirá la carga. Si omite este valor, la carga se escribe en la columna denominada `payload`.

Admite[Plantillas de sustitución \(p. 622\)](#): Sí

operation

(Opcional) El tipo de operación que se va a realizar. Valores válidos: INSERT, UPDATE, DELETE.

Admite[Plantillas de sustitución \(p. 622\)](#): Sí

roleARN

El rol de IAM que permite tener acceso a la tabla de DynamoDB. Para obtener más información, consulte [Requisitos \(p. 493\)](#).

Admite[Plantillas de sustitución \(p. 622\)](#): No

Los datos que se escriben en la tabla de DynamoDB son el resultado de la instrucción SQL de la regla.

Ejemplos

En el siguiente ejemplo de JSON se define una acción de DynamoDB en unAWS IoTregla.

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

Véase también

- [¿Qué es Amazon DynamoDB?](#)en laGuía para desarrolladores de Amazon DynamoDB
- [Introducción al DynamoDB](#)en laGuía para desarrolladores de Amazon DynamoDB
- [Tutorial: Almacenamiento de datos de dispositivos en una tabla de DynamoDB \(p. 209\)](#)

DynamoDBv2

El DynamoDBv2 (`dynamodbv2`) escribe todo un mensaje MQTT o parte de él en una tabla de Amazon DynamoDB. Cada atributo de la carga se escribe en una columna independiente de la base de datos de DynamoDB.

Requisitos

Esta acción de regla debe cumplir estos requisitos:

- Un rol de IAM queAWS IoT puede suponer que se lleva a cabo el `PutItem`. Para obtener más información, consulte [Concesión de unAWS IoT regla el acceso que requiere \(p. 473\)](#).

En el navegadorAWS IoT consola, puede elegir o crear un rol para permitirAWS IoT para llevar a cabo esta acción de regla de.

- La carga del mensaje MQTT debe contener una clave de nivel de raíz que coincida con la clave de partición principal de la tabla y una clave de nivel de raíz que coincide con la clave de ordenación principal de la tabla, si hay una definida.
- Si utilizas un cliente administradoAWS KMS key(clave KMS) para cifrar los datos en reposo de en DynamoDB, el servicio debe tener permiso para poder usar la clave KMS en nombre del autor de la llamada. Para obtener más información, consulta[Clave KMS administrada por el cliente](#)en laGuía de introducción a Amazon DynamoDB.

Parámetros

Al crear unAWS IoT con esta acción, debe especificar la información siguiente:

`PutItem`

Un objeto que especifica la tabla de DynamoDB en la que se escribirán los datos del mensaje. Este objeto debe incluir la siguiente información:

tableName

El nombre de la tabla de DynamoDB.

Admite[Plantillas de sustitución \(p. 622\)](#): API y AWS CLI sólo

roleARN

El rol de IAM que permite tener acceso a la tabla de DynamoDB. Para obtener más información, consulte [Requisitos \(p. 495\)](#).

Admite[Plantillas de sustitución \(p. 622\)](#): No

Los datos que se escriben en la tabla de DynamoDB son el resultado de la instrucción SQL de la regla.

Ejemplos

En el siguiente ejemplo de JSON se define una acción de DynamoDBv2 en unAWS IoTregla.

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

En el siguiente ejemplo de JSON se define una acción de DynamoDB con plantillas de sustitución en unAWS IoTregla.

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

Véase también

- [¿Qué es Amazon DynamoDB?](#)en laGuía para desarrolladores de Amazon DynamoDB

- [Introducción al DynamoDB](#) En la Guía para desarrolladores de Amazon DynamoDB

Elasticsearch

La Elasticsearch (`elasticsearch`) escribe datos de mensajes MQTT en Amazon OpenSearch Dominio de servicio. A continuación, puedes usar herramientas como OpenSearch Paneles para consultar y visualizar datos en OpenSearch Servicio

Warning

La `Elasticsearch` La acción solo la pueden utilizar las acciones de las reglas existentes. Para crear una nueva acción de regla o para actualizar una acción de regla existente, utilice el `OpenSearch` en su lugar de acción de regla. Para obtener más información, consulte [OpenSearch \(p. 538\)](#).

Requisitos

Esta acción de regla debe cumplir estos requisitos:

- Un rol de IAM que AWS IoT puede suponer que se lleva a cabo en el `:ESHttpPut`. Para obtener más información, consulte [Concesión de un AWS IoT regla el acceso que requiere \(p. 473\)](#).
En el navegador AWS IoT consola, puede elegir o crear un rol para permitir AWS IoT para llevar a cabo esta acción de regla de.
- Si utilizas un cliente administrado AWS KMS key (clave KMS) para cifrar los datos en reposo en reposo de OpenSearch, el servicio debe tener permiso para poder usar la clave KMS en nombre del autor de la llamada. Para obtener más información, consulta [Cifrado de datos en reposo para Amazon OpenSearch Service \(Servicio\)](#) en la Amazon OpenSearch Guía para desarrolladores de servicio.

Parámetros

Al crear un AWS IoT con esta acción, debe especificar la información siguiente:

`endpoint`

Punto de enlace de su dominio de servicio.

Admite [Plantillas de sustitución \(p. 622\)](#): API y AWS CLI sólo

`index`

Índice de donde se van a almacenar los datos.

Admite [Plantillas de sustitución \(p. 622\)](#): Sí

`type`

Tipo de documento que está almacenando.

Admite [Plantillas de sustitución \(p. 622\)](#): Sí

`id`

Identificador único de cada documento.

Admite [Plantillas de sustitución \(p. 622\)](#): Sí

`roleARN`

El rol de IAM que permite tener acceso a la OpenSearch Dominio de servicio. Para obtener más información, consulte [Requisitos \(p. 497\)](#).

Admite[Plantillas de sustitución \(p. 622\)](#): No

Ejemplos

En el siguiente ejemplo de JSON se define una acción Elasticsearch en unAWS IoTregla y cómo se pueden especificar los campos para laelasticsearchaction. Para obtener más información, consulte[ElasticsearchAction](#).

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

En el siguiente ejemplo de JSON se define una acción de Elasticsearch con plantillas de sustitución en unAWS IoTregla.

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

Véase también

- [OpenSearch \(p. 538\)](#)
- [¿Qué es Amazon? OpenSearch ¿Servicio de?](#)

HTTP

El HTTPS ([http](#)) envía datos de un mensaje MQTT a una aplicación o servicio web.

Requisitos

Esta acción de regla debe cumplir estos requisitos:

- Debe confirmar y habilitar los puntos de enlace HTTPS para que el motor de reglas pueda usarlo. Para obtener más información, consulte [Uso de destinos de reglas del tema HTTP \(p. 501\)](#).

Parámetros

Al crear unAWS IoTcon esta acción, debe especificar la información siguiente:

`url`

El punto de enlace HTTPS al que se envía el mensaje mediante el método HTTP POST. Si utiliza una dirección IP en lugar de un nombre de host, debe ser una dirección IPv4. No compatible con las direcciones IPv6.

Admite[Plantillas de sustitución \(p. 622\)](#): Sí

`confirmationUrl`

(Opcional) Si se especifica,AWS IoTutiliza la dirección URL de confirmación para crear un destino de regla del tema coincidente. Debe habilitar el destino de la regla del tema para poder usarlo en una acción HTTP. Para obtener más información, consulte [Uso de destinos de reglas del tema HTTP \(p. 501\)](#). Si utiliza plantillas de sustitución, debe crear manualmente destinos de reglas de tema para poder utilizar la acción `http. confirmationUrl` debe ser un prefijo de `url`.

La relación entre `url` y `confirmationUrl` se describe de la siguiente manera:

- Si `url` está codificado y no se proporciona `confirmationUrl`, tratamos implícitamente el campo `url` como `confirmationUrl`. AWS IoT crea un destino de regla del tema para `url`.
- Si `url` y `confirmationUrl` están codificados, `url` debe comenzar por `confirmationUrl`. AWS IoT crea un destino de regla del tema para `confirmationUrl`.
- Si `url` contiene una plantilla de sustitución, debe especificar `confirmationUrl` y `url` debe comenzar por `confirmationUrl`. Si `confirmationUrl` contiene plantillas de sustitución, debe crear manualmente destinos de reglas del tema para poder utilizar la acción `http`. Si `confirmationUrl` no contiene plantillas de sustitución, AWS IoT crea un destino de regla del tema para `confirmationUrl`.

Admite[Plantillas de sustitución \(p. 622\)](#): Sí

`headers`

(Opcional) Lista de encabezados que se van a incluir en las solicitudes HTTP al endpoint. Cada encabezado debe incluir la siguiente información:

`key`

La clave del encabezado.

Admite[Plantillas de sustitución \(p. 622\)](#): No

`value`

El valor del encabezado.

Admite[Plantillas de sustitución \(p. 622\)](#): Sí

`Note`

El tipo de contenido predeterminado es application/json cuando la carga útil está en formato JSON. De lo contrario, es application/octet-stream. Puede sobrescribirlo especificando el tipo

de contenido exacto en el encabezado con el tipo de contenido de la clave (sin distinción entre mayúsculas y minúsculas).

auth

(Opcional) Autenticación utilizada por el motor de reglas para conectarse a la dirección URL del punto de enlace especificada en el argumento `url`. Actualmente, Signature Version 4 es el único tipo de autenticación admitido. Para obtener más información, consulte [Autorización HTTP](#).

Admite [Plantillas de sustitución](#) (p. 622): No

Ejemplos

En el siguiente ejemplo de JSON se define un AWS IoT Rule con una acción 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 reintento de acción HTTP

La AWS IoT motor de reglas de reintenta la acción HTTP de acuerdo con estas reglas:

- El motor de reglas intenta enviar un mensaje al menos una vez.
- El motor de reglas realiza como máximo dos reintentos. El número máximo de intentos es tres.
- El motor de reglas no realiza un reintento si:
 - El intento anterior proporcionó una respuesta mayor de 16384 bytes.
 - El servicio o la aplicación web de salida cierra la conexión TCP después del intento.
 - El tiempo total para completar una solicitud con reintentos superó el límite de tiempo de espera de la solicitud.
 - La solicitud devuelve un código de estado HTTP distinto de 429, 500-599.

Note

Se aplican [costos estándar de transferencia de datos](#) a los reintentos.

Véase también

- [Uso de destinos de reglas del tema HTTP \(p. 501\)](#)
- [Ruta de datos directamente desde AWS IoT Core a sus servicios web en la Internet de las cosas en AWS blog](#)

Uso de destinos de reglas del tema HTTP

Un destino de regla de tema HTTP es un servicio web al que el motor de reglas puede enrutar datos desde una regla de tema. Una AWS IoT Core El recurso describe el servicio web de AWS IoT. Los recursos de destino de reglas de tema se pueden compartir mediante reglas diferentes.

Ante AWS IoT Core puede enviar datos a otro servicio web, debe confirmar que puede acceder al extremo del servicio.

Descripción del destino de la regla del tema HTTP

Un destino de regla de tema HTTP hace referencia a un servicio web que admite una URL de confirmación y una o más URL de recopilación de datos. El recurso de destino de la regla de temas HTTP contiene la URL de confirmación de su servicio web. Al configurar una acción de regla de tema HTTP, especifica la URL real del endpoint que debe recibir los datos junto con la URL de confirmación del servicio web. Una vez confirmado el destino, la regla de tema envía el resultado de la instrucción SQL al extremo HTTPS (y no a la URL de confirmación).

Un destino de regla de tema HTTP puede tener uno de los siguientes estados:

ENABLED

El destino se ha confirmado y se puede utilizar mediante una acción de regla. Un destino debe tener el estado ENABLED para que se utilice en una regla. Solo puede habilitar un destino con el estado DISABLED.

DISABLED

El destino se ha confirmado pero una acción de regla no lo puede utilizar. Esto es útil si desea impedir temporalmente el tráfico a su punto de enlace sin tener que pasar de nuevo por el proceso de confirmación. Solo puede deshabilitar un destino con el estado ENABLED.

IN_PROGRESS

La confirmación del destino se está realizando.

ERROR

Se ha agotado el tiempo de espera de confirmación del destino.

Una vez confirmado y habilitado el destino de una regla de tema HTTP, se puede utilizar con cualquier regla de su cuenta.

Las siguientes secciones describen acciones comunes en los destinos de reglas de temas HTTP.

Creación y confirmación de destinos de reglas de temas HTTP

Puede crear un destino de regla de tema HTTP llamando al `CreateTopicRuleDestination` mediante el uso de la AWS IoT consola de .

Después de crear un destino, AWS IoT envía una solicitud de confirmación a la URL de confirmación. La solicitud de confirmación tiene el siguiente 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"
}
```

El contenido de la solicitud de confirmación incluye la información siguiente:

arn

Nombre de recurso de Amazon (ARN) del destino de la regla del tema que se desea confirmar.

confirmationToken

El token de confirmación enviado por AWS IoT Core. El token del ejemplo está truncado. Su token será mayor. Necesitarás este token para confirmar tu destino con AWS IoT Core.

enableUrl

La dirección URL a la que se desplaza para confirmar el destino de una regla del tema.

messageType

Tipo de mensaje.

Para completar el proceso de confirmación de endpoint, debe realizar uno de estos procedimientos después de que la URL de confirmación reciba la solicitud de confirmación.

- Llame a `enableUrl` en la solicitud de confirmación y, a continuación, llame a `UpdateTopicRuleDestination` para establecer el estado de la regla del tema en `ENABLED`.
- Llame a `ConfirmTopicRuleDestination` operación y pasar el `confirmationToken` de la solicitud de confirmación.
- Copia `deconfirmationToken` y péguelo en el cuadro de diálogo de confirmación del destino en la AWS IoT consola de .

Envío de una nueva solicitud de confirmación

Para activar un nuevo mensaje de confirmación para un destino, llame a `UpdateTopicRuleDestination` y establezca el estado del destino de la regla del tema en `IN_PROGRESS`.

Tendrás que repetir el proceso de confirmación después de enviar una nueva solicitud de confirmación

Deshabilitación y eliminación de un destino de regla del tema

Para deshabilitar un destino, llame a `UpdateTopicRuleDestination` y establezca el estado del destino de la regla del tema en `DISABLED`. Se puede volver a activar una regla de tema en estado `DISABLED` sin necesidad de enviar una nueva solicitud de confirmación.

Para eliminar un destino de regla del tema, llame a `DeleteTopicRuleDestination`.

Autoridades de certificación admitidas por endpoints HTTPS en los destinos de reglas de tema

Los endpoints HTTPS admiten las siguientes autoridades de certificación en los destinos de reglas de tema.

```
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: utndatacorpsgcc  
Certificate fingerprints:  
MD5: B3:A5:3E:77:21:60: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: staatderlandenewrootca
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: atotrustedroot2011
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: soneraaclassica
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: cadisigroot1
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: certplusclass3ppprimaryca
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: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: soneraaclass2rootca
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: globalsecureccrootcar5
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:CO

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:CO

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: quo vadis root ca2g3
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: trustcorecal
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:CO: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:OB: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: secomsrootca2  
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: secomsrootca1
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: eecertificationcentreroottca
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: soneraaclass2ca
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:CO: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:CO: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:CO: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:BO: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: camerfirmachambersofcommerceroott  
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

El IoT Analytics (`iotAnalytics`) envía datos de un mensaje MQTT a unAWS IoT Analytics canal de.

Requisitos

Esta acción de regla debe cumplir estos requisitos:

- Un rol de IAM que AWS IoT puede suponer que se lleva a cabo el `iotanalytics:BatchPutMessage`. Para obtener más información, consulte [Concesión de unAWS IoT regla el acceso que requiere \(p. 473\)](#).

En el navegador AWS IoT consola, puede elegir o crear un rol para permitir AWS IoT para llevar a cabo esta acción de regla de.

La política asociada al rol que especifique debe tener el aspecto que se muestra en el siguiente ejemplo.

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

Parámetros

Al crear unAWS IoTcon esta acción, debe especificar la información siguiente:

batchMode

(Opcional) Si se debe procesar la acción como un lote. El valor predeterminado es **false**.

CuandobatchModeestrey la instrucción de SQL de regla se evalúa como una matriz, cada elemento de la matriz se entrega como un mensaje independiente cuando se pasa por**BatchPutMessage**alAWS IoTCanal de análisis. La matriz resultante no puede tener más de 100 mensajes.

Admite**Plantillas de sustitución** (p. 622): No

channelName

El nombre del canal de AWS IoT Analytics en el que se escriben los datos.

Admite**Plantillas de sustitución** (p. 622): API yAWS CLIsólo

roleArn

El rol de IAM que permite tener acceso a laAWS IoT Analyticscanal de. Para obtener más información, consulte [Requisitos](#) (p. 525).

Admite**Plantillas de sustitución** (p. 622): No

Ejemplos

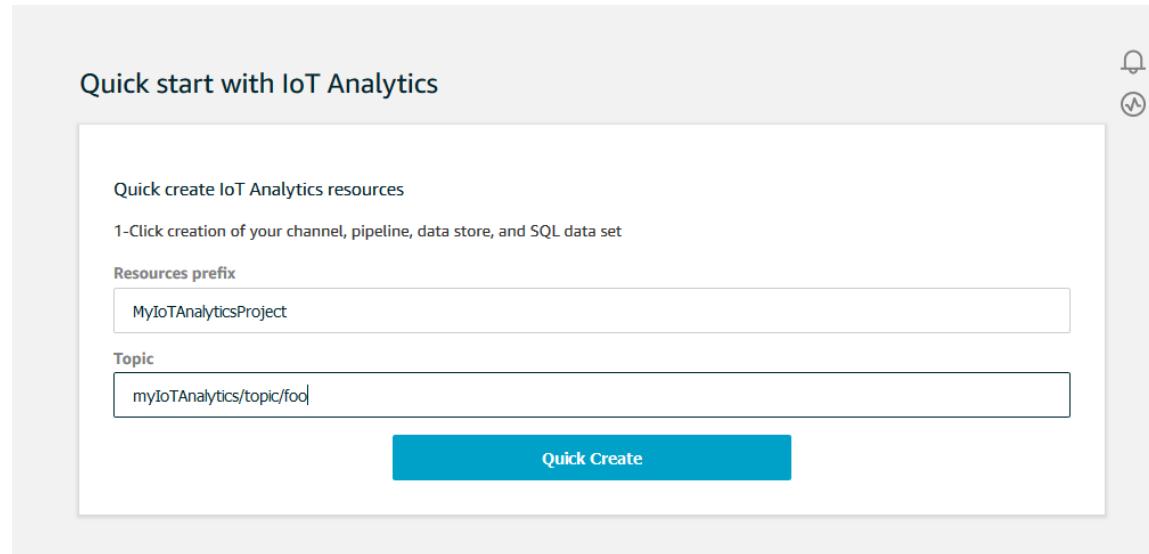
En el siguiente ejemplo de JSON se define una acción de IoT Analytics en unAWS IoTregla.

```
{  
    "topicRulePayload": {  
        "sql": "SELECT * FROM 'some/topic'",  
        "ruleDisabled": false,  
        "awsIotSqlVersion": "2016-03-23",  
        "actions": [  
            {  
                "iotAnalytics": {  
                    "channelName": "mychannel",  
                    "roleArn": "arn:aws:iam::123456789012:role/analyticsRole",  
                }  
            }  
        ]  
    }  
}
```

}

Véase también

- [¿Qué es AWS IoT Analytics?](#) en la Guía del usuario de AWS IoT Analytics
- La AWS IoT Analytics consola también tiene un inicio rápido que le permite crear un canal, almacén de datos, canalización y almacén de datos con un solo clic. Para obtener más información, consulte [AWS IoT Analytics Guía de inicio rápido de consola](#) en la AWS IoT Analytics Guía del usuario de.



IoT Events

Los IoT Events (`iotEvents`) envía datos de un mensaje MQTT a un AWS IoT Event entrada.

Requisitos

Esta acción de regla debe cumplir estos requisitos:

- Un rol de IAM que AWS IoT puede suponer que se lleva a cabo el `iotevents:BatchPutMessage`. Para obtener más información, consulte [Concesión de un AWS IoT regla el acceso que requiere \(p. 473\)](#).

En el navegador AWS IoT consola, puede elegir o crear un rol para permitir AWS IoT para llevar a cabo esta acción de regla de.

Parámetros

Al crear un AWS IoT con esta acción, debe especificar la información siguiente:

`batchMode`

(Opcional) Si se procesan las acciones de evento como un lote. El valor predeterminado es `false`.

Cuando `batchMode` es `true` la instrucción de SQL de regla se evalúa como una matriz, cada elemento de la matriz se trata como un mensaje independiente cuando se envía a AWS IoT Events llamando `BatchPutMessage`. La matriz resultante no puede tener más de 10 mensajes.

Cuando `batchMode` es `true`, no puede especificar un `messageId`.

Admite[Plantillas de sustitución \(p. 622\)](#): No

`inputName`

El nombre de la entrada de AWS IoT Events.

Admite[Plantillas de sustitución \(p. 622\)](#): API y AWS CLI sólo

`messageId`

(Opcional) Utilice esta opción para asegurarse de que solo una entrada (mensaje) con un determinado `messageId` se procesa por un AWS IoT Events detector de. Puede utilizar el `el${newuuid()}` plantilla de sustitución para generar un ID exclusivo para cada solicitud.

Cuando `batchMode` es `true`, no puede especificar un `messageId`. Se asignará un nuevo valor UUID.

Admite[Plantillas de sustitución \(p. 622\)](#): Sí

`roleArn`

El rol de IAM que permite AWS IoT para enviar una entrada a AWS IoT Events detector de. Para obtener más información, consulte [Requisitos \(p. 527\)](#).

Admite[Plantillas de sustitución \(p. 622\)](#): No

Ejemplos

En el siguiente ejemplo de JSON se define una acción de IoT Events en un AWS IoT regla.

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

Véase también

- [¿Qué es AWS IoT Events?](#) en la AWS IoT Events Guía para desarrolladores

IoT SiteWise

El IoT SiteWise (`iotSiteWise`) envía datos de un mensaje MQTT a las propiedades de recurso en AWS IoT SiteWise.

Puede seguir un tutorial que le muestra cómo incorporar datos de objetos de AWS IoT. Para obtener más información, consulte [Incorporación de datos en AWS IoT SiteWise desde AWS IoT Things](#) en la [AWS IoT SiteWise Guía del usuario](#).

Requisitos

Esta acción de regla debe cumplir estos requisitos:

- Un rol de IAM que AWS IoT puede suponer que se lleva a cabo `iotsitewise:BatchPutAssetPropertyValue`. Para obtener más información, consulte [Concesión de un AWS IoT Regla el acceso que requiere \(p. 473\)](#).

Puede asociar el siguiente ejemplo de política de confianza al rol.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": "iotsitewise:BatchPutAssetPropertyValue",  
            "Resource": "*"  
        }  
    ]  
}
```

Para mejorar la seguridad, puede especificar una ruta jerárquica de recursos de AWS IoT SiteWise en la propiedad `Condition`. El siguiente ejemplo es una política de confianza que especifica una ruta jerárquica de recursos.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": "iotsitewise:BatchPutAssetPropertyValue",  
            "Resource": "*",  
            "Condition": {  
                "StringLike": {  
                    "iotsitewise:assetHierarchyPath": [  
                        "/root node asset ID",  
                        "/root node asset ID/*"  
                    ]  
                }  
            }  
        }  
    ]  
}
```

- Cuando envías datos a AWS IoT SiteWise con esta acción, los datos deben cumplir los requisitos del `BatchPutAssetPropertyValue`. Para obtener más información, consulte [BatchPutAssetPropertyValue](#) en la Referencia de la API AWS IoT SiteWise.

Parámetros

Al crear un AWS IoT con esta acción, debe especificar la información siguiente:

`putAssetPropertyValueEntries`

Una lista de entradas de valor de propiedad de recurso, en la que cada entrada contiene la siguiente información:

propertyAlias

(Opcional) El alias de propiedad asociado a la propiedad del recurso. Debe especificar un `propertyAlias` o tanto un `assetId` como un `propertyId`. Para obtener más información acerca de alias de propiedades, consulte [Asignación de flujos de datos industriales a propiedades de activos](#) en la AWS IoT SiteWise Guía del usuario de.

Admite [Plantillas de sustitución \(p. 622\)](#): Sí

assetId

(Opcional) ID del AWS IoT SiteWise recurso de. Debe especificar un `propertyAlias` o tanto un `assetId` como un `propertyId`.

Admite [Plantillas de sustitución \(p. 622\)](#): Sí

propertyId

(Opcional) ID de la propiedad del activo. Debe especificar un `propertyAlias` o tanto un `assetId` como un `propertyId`.

Admite [Plantillas de sustitución \(p. 622\)](#): API y AWS CLI sólo

entryId

(Opcional) Identificador único para esta entrada. Puede definir el `entryId` para facilitar el seguimiento del mensaje que causó un error en caso de que se produzca un error. El valor predeterminado es un nuevo UUID.

Admite [Plantillas de sustitución \(p. 622\)](#): Sí

propertyValues

Una lista de valores de propiedad que se van a insertar, en la que cada valor contiene una marca temporal, la calidad y el valor (TQV) en el formato siguiente:

timestamp

Una estructura de marca temporal que contiene la siguiente información:

timeInSeconds

Una cadena que contiene el tiempo en segundos en formato de tiempo Unix. Si su carga de mensajes no tiene una marca temporal, puede usar [timestamp\(\) \(p. 613\)](#), que devuelve la hora actual en milisegundos. Para convertir ese tiempo en segundos, puede utilizar la siguiente plantilla de sustitución: `${\lfloor timestamp() / 1E3 \rfloor}`.

Admite [Plantillas de sustitución \(p. 622\)](#): Sí

offsetInNanos

(Opcional) Una cadena que contiene el desfase de tiempo en nanosegundos con respecto al tiempo en segundos. Si su carga de mensajes no tiene una marca temporal, puede usar [timestamp\(\) \(p. 613\)](#), que devuelve la hora actual en milisegundos. Para calcular el desfase en nanosegundos a partir de ese tiempo, puede utilizar la siguiente plantilla de sustitución: `${(timestamp() % 1E3) * 1E6}`.

Admite [Plantillas de sustitución \(p. 622\)](#): Sí

Con respecto a la época de Unix, AWS IoT SiteWise Solo acepta entradas que tengan una marca de tiempo de hasta 7 días en el pasado y hasta 5 minutos en el futuro.

quality

(Opcional) Una cadena que describe la calidad del valor. Valores válidos: GOOD, BAD, UNCERTAIN.

Admite [Plantillas de sustitución \(p. 622\)](#): Sí

value

Una estructura de valores que contiene uno de los siguientes campos de valor, en función del tipo de datos de la propiedad del recurso:

booleanValue

(Opcional) Una cadena que contiene el valor booleano de la entrada del valor.

Admite[Plantillas de sustitución \(p. 622\)](#): Sí

doubleValue

(Opcional) Una cadena que contiene el valor «double» de la entrada del valor.

Admite[Plantillas de sustitución \(p. 622\)](#): Sí

integerValue

(Opcional) Una cadena que contiene el valor entero de la entrada del valor.

Admite[Plantillas de sustitución \(p. 622\)](#): Sí

stringValue

(Opcional) El valor de cadena de la entrada del valor.

Admite[Plantillas de sustitución \(p. 622\)](#): Sí

roleArn

El ARN del rol de IAM que concede AWS IoT permiso para enviar un valor de propiedad de activo a AWS IoT SiteWise. Para obtener más información, consulte [Requisitos \(p. 529\)](#).

Admite[Plantillas de sustitución \(p. 622\)](#): No

Ejemplos

En el siguiente ejemplo de JSON se define un IoT básico SiteWise acción en un AWS IoT regla.

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

```
        ]
    }
}
```

En el siguiente ejemplo de JSON se define un IoT SiteWise acción en unAWS IoTRegla. En este ejemplo se utiliza el tema como alias de propiedad y función `timestamp()`. Por ejemplo, si publica datos en `/company/windfarm/3/turbine/7/rpm`, esta acción envía los datos a la propiedad del recurso con un alias de propiedad que es el mismo que el tema 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"
                        }
                    ]
                }
            }
        ]
    }
}
```

Véase también

- [¿Qué es AWS IoT SiteWise?](#) en la [Guía del usuario de AWS IoT SiteWise](#)
- [Incorporación de datos medianteAWS IoT CoreReglas de](#)en la [AWS IoT SiteWiseGuía del usuario de](#)
- [Incorporación de datos enAWS IoT SiteWisedesdeAWS IoTthingsen](#) la [AWS IoT SiteWiseGuía del usuario de](#)
- [Solución de problemas deAWS IoT SiteWiseAcción de la regla de](#)en la [AWS IoT SiteWiseGuía del usuario de](#)

Kinesis Data Firehose

La manguera de Kinesis Data Firehose (`firehose`) envía datos de un mensaje MQTT a un flujo de Amazon Kinesis Data Firehose.

Requisitos

Esta acción de regla debe cumplir estos requisitos:

- Un rol de IAM que AWS IoT puede suponer que se lleva a cabo el `firehose:PutRecord`. Para obtener más información, consulte [Concesión de un AWS IoT regla el acceso que requiere \(p. 473\)](#).

En el navegador AWS IoT consola, puede elegir o crear un rol para permitir AWS IoT para llevar a cabo esta acción de regla de.

- Si utiliza Kinesis Data Firehose para enviar datos a un bucket de Amazon S3 y utiliza un AWS Key Management Service(AWS KMS) administradas por el cliente AWS KMS key(clave KMS) para los cifrar datos en reposo de Amazon S3, Kinesis Data Firehose debe tener acceso al bucket y permiso para poder usar la AWS KMS key en nombre de la persona que llama. Para obtener más información, consulte [Concesión de acceso de Kinesis Data Firehose a un destino de Amazon S3](#) en la Amazon Kinesis Data Firehose Developer Guide.

Parámetros

Al crear un AWS IoT con esta acción, debe especificar la información siguiente:

`batchMode`

(Opcional) Para entregar el flujo de Kinesis Data Firehose como un lote mediante el uso de `PutRecordBatch`. El valor predeterminado es `false`.

Cuando `batchMode` es `true` y la instrucción SQL de la regla se evalúa como una matriz, cada elemento de la matriz forma un registro en el `PutRecordBatch` request. La matriz resultante no puede tener más de 500 registros.

Admite [Plantillas de sustitución \(p. 622\)](#): No

`deliveryStreamName`

El flujo de Kinesis Data Firehose en el que deben escribirse los datos del mensaje.

Admite [Plantillas de sustitución \(p. 622\)](#): API y AWS CLIsólo

`separator`

(Opcional) Un separador de caracteres que se utilizará para separar registros escritos en el flujo de Kinesis Data Firehose. Si omite este parámetro, el flujo no utiliza ningún separador. Valores válidos: `,` (coma), `\t` (pestaña), `\n` (línea nueva), `\r\n` (Nueva línea de Windows).

Admite [Plantillas de sustitución \(p. 622\)](#): No

`roleArn`

El rol de IAM que permite el acceso al flujo de Kinesis Data Firehose. Para obtener más información, consulte [Requisitos \(p. 532\)](#).

Admite [Plantillas de sustitución \(p. 622\)](#): No

Ejemplos

En el siguiente ejemplo de JSON se define una acción de Kinesis Data Firehose en un AWS IoT regla.

```
{  
    "topicRulePayload": {  
        "sql": "SELECT * FROM 'some/topic'",  
        "ruleDisabled": false,  
        "awsIotSqlVersion": "2016-03-23",  
        "actions": [  
            {  
                "firehose": {  
                    "deliveryStreamName": "my_firehose_stream",  
                    "region": "us-east-1"  
                }  
            }  
        ]  
    }  
}
```

```
        "roleArn": "arn:aws:iam::123456789012:role/aws_iot_firehose"
    }
}
}
```

En el siguiente ejemplo de JSON se define una acción de Kinesis Data Firehose con plantillas de sustitución en unAWS IoTregla.

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

Véase también

- [¿Qué es Amazon Kinesis Data Firehose?](#)en laAmazon Kinesis Data Firehose Developer Guide

Kinesis Data Streams

Kinesis Data Streams (`kinesis`) escribe datos de un mensaje MQTT en Amazon Kinesis Data Streams.

Requisitos

Esta acción de regla debe cumplir estos requisitos:

- Un rol de IAM queAWS IoT puede suponer que se lleva a cabo el`kinesis:PutRecord`. Para obtener más información, consulte [Concesión de unAWS IoTregla el acceso que requiere \(p. 473\)](#).

En el navegadorAWS IoTconsola, puede elegir o crear un rol para permitirAWS IoTpara llevar a cabo esta acción de regla de.

- Si usa unAWS Key Management Service(AWS KMS) administradas por el clienteAWS KMS key(clave KMS) para cifrar los datos en reposo en reposo de en Kinesis Data Streams, el servicio debe tener permiso para poder usar elAWS KMS keyen nombre de la persona que llama. Para obtener más información, consulte[Permisos para utilizar generados por el usuarioAWS KMS keysen laAmazon Kinesis Data Streams Developer Guide](#).

Parámetros

Al crear unAWS IoTcon esta acción, debe especificar la información siguiente:

`stream`

El flujo de datos de Kinesis en el que se escriben los datos.

Admite[Plantillas de sustitución \(p. 622\)](#): API yAWS CLIsólo
`partitionKey`

La clave de partición utilizada para determinar en qué fragmento se escriben los datos. La clave de partición suele estar compuesta por una expresión (por ejemplo,`#{topic()}o#{timestamp()}`).

Admite[Plantillas de sustitución \(p. 622\)](#): Sí
`roleArn`

El ARN del rol de IAM que concedeAWS IoTpermiso para acceder a la secuencia de datos de Kinesis. Para obtener más información, consulte [Requisitos \(p. 534\)](#).

Admite[Plantillas de sustitución \(p. 622\)](#): No

Ejemplos

En el siguiente ejemplo de JSON se define una acción de Kinesis Data Streams en unAWS IoTregla.

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

En el siguiente ejemplo de JSON se define una acción de Kinesis con plantillas de sustitución en unAWS IoTregla.

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

Véase también

- [¿Qué es Amazon Kinesis Data Streams?](#)en laAmazon Kinesis Data Streams Developer Guide

Lambda

`lambda:lambda`) invoca unAWS Lambda, pasando un mensaje de MQTT.AWS IoTinvoca funciones de Lambda de forma asíncrona.

Puede seguir un tutorial que le muestra cómo crear y probar una regla con una acción Lambda. Para obtener más información, consulte [Tutorial: Dar formato a una notificación mediante unAWS Lambdafunción \(p. 214\)](#).

Requisitos

Esta acción de regla debe cumplir estos requisitos:

- ParaAWS IoTpara invocar una función Lambda, debe configurar una política que conceda `lambda:InvokeFunction`permiso paraAWS IoT. Solo puede invocar una función Lambda definida en la mismaRegión de AWSdonde existe su política Lambda. Las funciones de Lambda utilizan políticas basadas en recursos, por lo que debe asociar la política a la función Lambda en sí.

Utilice lo siguienteAWS CLIpaa asociar una política que conceda `lambda:InvokeFunction`permiso.

```
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"
```

Laadd-permissionespera los parámetros siguientes:

--function-name

Nombre de la función Lambda. Agrega un nuevo permiso para actualizar la política de recursos de la función.

--region

LaRegión de AWSde la función.

--principal

El director que recibe el permiso. Esto debería ser*iot.amazonaws.com*para permitirAWS IoTpermiso para llamar a la función Lambda.

--source-arn

El ARN de la regla. Puede utilizar elget-topic-rule AWS CLIpaa obtener el ARN de una regla.

--source-account

LaCuenta de AWSdonde está definida la regla de.

--statement-id

Un identificador de instrucción único.

--action

La acción Lambda que desea permitir en esta instrucción. Para permitirAWS IoTpara invocar una función Lambda, especifique`lambda:InvokeFunction`.

Important

Si agrega un permiso para unAWS IoTprincipal sin proporcionar elsource-arnsource-account, cualquierCuenta de AWSque crea una regla con su acción Lambda puede activar reglas para invocar su función Lambda desdeAWS IoT.

Para obtener más información, consulte [Permisos de AWS Lambda](#).

- Si usa un AWS Key Management Service(AWS KMS) administradas por el clienteAWS KMS key(clave KMS) para cifrar los datos en reposo en Lambda, el servicio debe tener permiso para poder usar el AWS KMS key en nombre de la persona que llama. Para obtener más información, consulte [Cifrado en reposo](#) en la AWS Lambda Guía para desarrolladores.

Parámetros

Al crear un AWS IoT con esta acción, debe especificar la información siguiente:

functionArn

El ARN de la función de Lambda que se va a invocar. AWS IoT debe tener permiso para invocar la función. Para obtener más información, consulte [Requisitos \(p. 536\)](#).

Si no especifica una versión o alias para la función de Lambda, se ejecutará la versión más reciente de la función. Puede especificar una versión o alias si desea ejecutar una versión específica de su función Lambda. Para especificar una versión o alias, añada la versión o el alias en el ARN de la función Lambda.

```
arn:aws:lambda:us-east-2:123456789012:function:myLambdaFunction:someAlias
```

Para obtener más información acerca del control de versiones y los alias, consulte [AWS Lambda versiones y alias de funciones](#).

Admite [Plantillas de sustitución \(p. 622\)](#): API y AWS CLI sólo

Ejemplos

En el siguiente ejemplo de JSON se define una acción Lambda en un AWS IoT regla.

```
{
    "topicRulePayload": {
        "sql": "SELECT * FROM 'some/topic'",
        "ruleDisabled": false,
        "awsIotSqlVersion": "2016-03-23",
        "actions": [
            {
                "lambda": {
                    "functionArn": "arn:aws:lambda:us-east-2:123456789012:function:myLambdaFunction"
                }
            }
        ]
    }
}
```

En el siguiente ejemplo de JSON se define una acción Lambda con plantillas de sustitución en un AWS IoT regla.

```
{
    "topicRulePayload": {
        "sql": "SELECT * FROM 'some/topic'",
        "ruleDisabled": false,
        "awsIotSqlVersion": "2016-03-23",
        "actions": [
            {
                "lambda": {

```

```
        "functionArn": "arn:aws:lambda:us-east-1:123456789012:function:  
${topic()}"  
    }  
}  
]  
}  
}
```

Véase también

- [¿Qué es AWS Lambda?](#)en laAWS LambdaGuía para desarrolladores
- Tutorial: Dar formato a una notificación mediante unAWS Lambdafunción (p. 214)

OpenSearch

La OpenSearch (`openSearch`) escribe datos de mensajes MQTT en Amazon OpenSearch Dominio de servicio. A continuación, puedes usar herramientas como OpenSearch Paneles para consultar y visualizar datos en OpenSearch Servicio

Requisitos

Esta acción de regla debe cumplir estos requisitos:

- Un rol de IAM queAWS IoT puede suponer que se lleva a cabo el `:ESHttpPut`. Para obtener más información, consulte [Concesión de unAWS IoTregla el acceso que requiere](#) (p. 473).

En el navegadorAWS IoTconsola, puedes elegir o crear un rol para permitirAWS IoTpara llevar a cabo esta acción de regla de.

- Si utilizas un cliente administradoAWS KMS key(clave KMS) para cifrar los datos en reposo en OpenSearch Servicio, el servicio debe tener permiso para poder usar la clave KMS en nombre del autor de la llamada. Para obtener más información, consulte[Cifrado de datos en reposo para Amazon OpenSearch Service \(Servicio\)](#)en laAmazon OpenSearch Guía para desarrolladores de servicio.

Parámetros

Al crear unAWS IoTcon esta acción, debe especificar la información siguiente:

`endpoint`

El punto final de tu Amazon OpenSearch Dominio de servicio.

Admite[Plantillas de sustitución](#) (p. 622): API yAWS CLIsólo

`index`

La OpenSearch índice donde deseas almacenar los datos.

Admite[Plantillas de sustitución](#) (p. 622): Sí

`type`

Tipo de documento que estás almacenando.

Admite[Plantillas de sustitución](#) (p. 622): Sí

`id`

Identificador único de cada documento.

Admite[Plantillas de sustitución \(p. 622\)](#): Sí
roleARN

El rol de IAM que permite tener acceso a la OpenSearch Dominio de servicio. Para obtener más información, consulte [Requisitos \(p. 538\)](#).

Admite[Plantillas de sustitución \(p. 622\)](#): No

Limitaciones

La OpenSearch (`openSearch`) no se puede utilizar para entregar datos a los clústeres de Elasticsearch de VPC.

Ejemplos

En el siguiente ejemplo de JSON se define un OpenSearch acción en unAWS IoTregla y cómo se pueden especificar los campos para laOpenSearchaction. Para obtener más información, consulte[Acción Abrir búsqueda](#).

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

En el siguiente ejemplo de JSON se define un OpenSearch acción con plantillas de sustitución en unAWS IoTregla.

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

Véase también

[¿Qué es Amazon OpenSearch?](#) [Servicio de?](#) en la [Amazon OpenSearch Guía para desarrolladores de servicio](#)

Republish

La republicación (`republish`) publica de nuevo un mensaje MQTT en otro tema MQTT.

Requisitos

Esta acción de regla debe cumplir estos requisitos:

- Un rol de IAM que AWS IoT puede suponer que se lleva a cabo el `:Publish`. Para obtener más información, consulte [Concesión de un AWS IoT regla el acceso que requiere \(p. 473\)](#).

En el navegador AWS IoT consola, puede elegir o crear un rol para permitir AWS IoT para llevar a cabo esta acción de regla de.

Parámetros

Al crear un AWS IoT con esta acción, debe especificar la información siguiente:

`topic`

El tema MQTT en el que se va a volver a publicar el mensaje.

Para volver a publicar en un tema reservado, que comienza con \$, use \$\$ en lugar de. Por ejemplo, para volver a publicar en el tema de sombra del dispositivo \$aws/things/MyThing/shadow/update, especifique el tema como \$\$aws/things/MyThing/shadow/update.

Note

Republishing to [Temas de trabajos reservados \(p. 107\)](#) no es compatible.

Admite [Plantillas de sustitución \(p. 622\)](#): Sí

`qos`

(Opcional) El nivel de calidad de servicio (QoS) que se utiliza para volver a publicar mensajes. Valores válidos: 0, 1. El valor predeterminado es 0. Para obtener más información acerca de QoS MQTT, consulte [MQTT \(p. 85\)](#).

Admite [Plantillas de sustitución \(p. 622\)](#): No

`roleArn`

El rol de IAM que permite AWS IoT para publicar en el tema de MQTT. Para obtener más información, consulte [Requisitos \(p. 540\)](#).

Admite [Plantillas de sustitución \(p. 622\)](#): No

Ejemplos

En el siguiente ejemplo de JSON se define una acción de republicación en un AWS IoT regla.

```
{
```

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

En el siguiente ejemplo de JSON se define una acción de republicación con plantillas de sustitución en unAWS IoTregla.

```
{  
    "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

El S3 (`s3`) escribe los datos de un mensaje MQTT en un bucket de Amazon Simple Storage Service (Amazon S3).

Requisitos

Esta acción de regla debe cumplir estos requisitos:

- Un rol de IAM queAWS IoTpuede suponer que se lleva a cabo el `s3:PutObject`. Para obtener más información, consulte [Concesión de unAWS IoTregla el acceso que requiere \(p. 473\)](#).

En el navegadorAWS IoTconsola, puede elegir o crear un rol para permitirAWS IoTpara llevar a cabo esta acción de regla de.

- Si usa unAWS Key Management Service(AWS KMS) administradas por el clienteAWS KMS key(clave KMS) para cifrar los datos en reposo de Amazon S3, el servicio debe tener permiso para poder usar elAWS KMS keyen nombre de la persona que llama. Para obtener más información, consulte [AWS administradoAWS KMS keysy administrado por el clienteAWS KMS keysen laAmazon Simple Storage Service Developer Guide](#).

Parámetros

Al crear unAWS IoTcon esta acción, debe especificar la información siguiente:

bucket

El bucket de Amazon S3 en el que se escriben los datos.

Admite[Plantillas de sustitución \(p. 622\)](#): API y AWS CLIsólo
`cannedacl`

(Opcional) La lista de control de acceso (ACL) predefinida de Amazon S3 que controla el acceso al objeto identificado mediante la clave de objeto. Para obtener más información, incluidos los valores permitidos, consulte[ACL predefinidas](#).

Admite[Plantillas de sustitución \(p. 622\)](#): No

key

La ruta al archivo en el que se escriben los datos.

Considere un ejemplo en el que se encuentra este parámetro `${topic()}/${timestamp()}`y la regla recibe un mensaje en el que el tema `essome/topic`. Si la marca de hora actual es`1460685389`, a continuación, esta acción escribe los datos en un archivo llamado`1460685389en` la carpeta `topic` del bucket de S3.

Note

Si utilizas una clave estática,AWS IoT sobrescribe un solo archivo cada vez que se invoca la regla. Le recomendamos que utilice la marca de tiempo del mensaje u otro identificador de mensaje único, de modo que por cada mensaje recibido se guarde un archivo nuevo en Amazon S3.

Admite[Plantillas de sustitución \(p. 622\)](#): Sí

roleArn

El rol de IAM que permite tener acceso al bucket de Amazon S3. Para obtener más información, consulte[Requisitos \(p. 541\)](#).

Admite[Plantillas de sustitución \(p. 622\)](#): No

Ejemplos

En el siguiente ejemplo de JSON se define una acción de S3 en unAWS IoTregla.

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

Véase también

- [¿Qué es Amazon S3?](#)en laGuía del usuario de Amazon Simple Storage Service

Salesforce IoT IoT

El IoT de Salesforce (salesforce) envía datos del mensaje MQTT que activó la regla a un flujo de entrada de Salesforce IoT.

Parámetros

Al crear unAWS IoTcon esta acción, debe especificar la información siguiente:

url

La dirección URL expuesta por el flujo de entrada de Salesforce IoT. La dirección URL está disponible en la plataforma de Salesforce IoT cuando crea un flujo de entrada. Para obtener más información, consulte la documentación de Salesforce IoT.

Admite[Plantillas de sustitución \(p. 622\)](#): No

token

El token que se utiliza para autenticar el acceso al flujo de entrada de Salesforce IoT especificado. El token está disponible en la plataforma de Salesforce IoT cuando se crea un flujo de entrada. Para obtener más información, consulte la documentación de Salesforce IoT.

Admite[Plantillas de sustitución \(p. 622\)](#): No

Ejemplos

En el siguiente ejemplo de JSON se define una acción IoT de Salesforce en unAWS IoTregla.

```
{  
    "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

El SNS (sns) envía los datos de un mensaje MQTT como notificación push de Amazon Simple Notification Service (Amazon SNS)

Puede seguir un tutorial que le muestra cómo crear y probar una regla con una acción SNS. Para obtener más información, consulte [Tutorial: Envío de una notificación de Amazon SNS \(p. 201\)](#).

Note

La acción SNS no admite [Temas de Amazon SNS FIFO \(First-In-First-In-First-In\)](#). Dado que el motor de reglas es un servicio totalmente distribuido, no se garantiza el orden de los mensajes cuando se invoca la acción de SNS.

Requisitos

Esta acción de regla debe cumplir estos requisitos:

- Un rol de IAM que AWS IoT puede suponer que se lleva a cabo el sns : Publish. Para obtener más información, consulte [Concesión de un AWS IoT regla el acceso que requiere \(p. 473\)](#).

En el navegador AWS IoT consola, puede elegir o crear un rol para permitir AWS IoT para llevar a cabo esta acción de regla de.

- Si usa un AWS Key Management Service(AWS KMS) administradas por el cliente AWS KMS key(clave KMS) para cifrar los datos en reposo en reposo de Amazon SNS, el servicio debe tener permiso para poder usar el AWS KMS key en nombre de la persona que llama. Para obtener más información, consulte [Administración de claves](#)en la Guía para desarrolladores de Amazon Simple Notification Service.

Parámetros

Al crear un AWS IoT con esta acción, debe especificar la información siguiente:

targetArn

El tema de SNS o el dispositivo individual al que se enviará la notificación de inserción.

Admite [Plantillas de sustitución \(p. 622\)](#): API y AWS CLIs sólo

messageFormat

(Opcional) El formato del mensaje. Amazon SNS utiliza esta configuración para determinar si la carga debe analizarse y si las partes pertinentes de la carga específicas de la plataforma deben extraerse. Valores válidos: JSON, RAW. El valor predeterminado es RAW.

Admite [Plantillas de sustitución \(p. 622\)](#): No

roleArn

El rol de IAM que permite tener acceso a SNS. Para obtener más información, consulte [Requisitos \(p. 544\)](#).

Admite [Plantillas de sustitución \(p. 622\)](#): No

Ejemplos

En el siguiente ejemplo de JSON se define una acción SNS en un AWS IoT regla.

```
{  
    "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",  
                    "messageFormat": "RAW"  
                }  
            }  
        ]  
    }  
}
```

```
        "roleArn": "arn:aws:iam::123456789012:role/aws_iot_sns"
    }
}
}
```

En el siguiente ejemplo de JSON se define una acción SNS con plantillas de sustitución en unAWS IoTregla.

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

Véase también

- [¿En qué consiste Amazon Simple Notification Service?](#)en laGuía para desarrolladores de Amazon Simple Notification Service
- [Tutorial: Envío de una notificación de Amazon SNS \(p. 201\)](#)

SQS

SQS (sqss) envía los datos de un mensaje MQTT a una cola de Amazon Simple Queue Service (Amazon SQS)

Note

La acción SQS no admite[Colas FIFO \(First-In-First-In-First-In-First-In-\)](#). Dado que el motor de reglas es un servicio totalmente distribuido, no se garantiza el orden de los mensajes cuando se activa la acción de SQS.

Requisitos

Esta acción de regla debe cumplir estos requisitos:

- Un rol de IAM queAWS IoT puede suponer que se lleva a cabo el`sqss:SendMessage`. Para obtener más información, consulte [Concesión de unAWS IoTregla el acceso que requiere \(p. 473\)](#).

En el navegadorAWS IoTconsola, puede elegir o crear un rol para permitirAWS IoTpara llevar a cabo esta acción de regla de.

- Si usa unaAWS Key Management Service(AWS KMS) administradas por el clienteAWS KMS key(clave KMS) para cifrar los datos en reposo de Amazon SQS, el servicio debe tener permiso para poder usar elAWS KMS keyen nombre de la persona que llama. Para obtener más información, consulte[Administración de claves](#)en laAmazon Simple Queue Service Developer Guide.

Parámetros

Al crear unAWS IoT con esta acción, debe especificar la información siguiente:

queueUrl

La URL de la cola de Amazon SQS en la que se escriben los datos.

Admite[Plantillas de sustitución \(p. 622\)](#): API y AWS CLIsólo

useBase64

Establezca este parámetro en `true`para configurar la acción de regla para codificar en base64 los datos del mensaje antes de escribir los datos en la cola de Amazon SQS. El valor predeterminado es `false`.

Admite[Plantillas de sustitución \(p. 622\)](#): No

roleArn

El rol de IAM que permite tener acceso a la cola de Amazon SQS. Para obtener más información, consulte [Requisitos \(p. 545\)](#).

Admite[Plantillas de sustitución \(p. 622\)](#): No

Ejemplos

En el siguiente ejemplo de JSON se define una acción SQS en unAWS IoTregla.

```
{  
    "topicRulePayload": {  
        "sql": "SELECT * FROM 'some/topic'",  
        "ruleDisabled": false,  
        "awsIotSqlVersion": "2016-03-23",  
        "actions": [  
            {  
                "sq": {  
                    "queueUrl": "https://sqs.us-east-2.amazonaws.com/123456789012/  
my_sq_queue",  
                    "roleArn": "arn:aws:iam::123456789012:role/aws_iot_sq"  
                }  
            }  
        ]  
    }  
}
```

En el siguiente ejemplo de JSON se define una acción SQS con plantillas de sustitución en unAWS IoTregla.

```
{  
    "topicRulePayload": {  
        "sql": "SELECT * FROM 'some/topic'",  
        "ruleDisabled": false,  
        "awsIotSqlVersion": "2016-03-23",  
        "actions": [  
            {  
                "sq": {  
                    "queueUrl": "https://sqs.us-east-2.amazonaws.com/123456789012/  
${topic()}",  
                    "useBase64": true,  
                    "roleArn": "arn:aws:iam::123456789012:role/aws_iot_sq"  
                }  
            }  
        ]  
    }  
}
```

```
        }
    ]
}
}
```

Véase también

- [¿Qué es Amazon Simple Queue Service?](#) en la Guía para desarrolladores de Amazon Simple Queue Service

Step Functions

Las funciones del paso `descheduleFunction` inicia unAWS Step Functions máquina de estado de.

Requisitos

Esta acción de regla debe cumplir estos requisitos:

- Un rol de IAM que AWS IoT puede suponer que se lleva a cabo el `states:StartExecution`. Para obtener más información, consulte [Concesión de unAWS IoT regla el acceso que requiere \(p. 473\)](#).

En el navegador AWS IoT consola, puede elegir o crear un rol para permitir AWS IoT para llevar a cabo esta acción de regla de.

Parámetros

Al crear unAWS IoT con esta acción, debe especificar la información siguiente:

`stateMachineName`

El nombre del equipo de estado de Step Functions que se va a iniciar.

Admite [Plantillas de sustitución \(p. 622\)](#): API y AWS CLIsólo

`executionNamePrefix`

(Opcional) El nombre dado a la ejecución de la máquina de estado consiste en este prefijo seguido de un UUID. Si no se proporciona uno, Step Functions crea un nombre exclusivo para cada ejecución de la máquina de estado.

Admite [Plantillas de sustitución \(p. 622\)](#): Sí

`roleArn`

El ARN del rol de que concede AWS IoT permiso para iniciar el equipo de estado. Para obtener más información, consulte [Requisitos \(p. 547\)](#).

Admite [Plantillas de sustitución \(p. 622\)](#): No

Ejemplos

En el siguiente ejemplo de JSON se define una acción Step Functions en unAWS IoT regla.

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

Véase también

- [¿Qué es AWS Step Functions?](#)en laAWS Step FunctionsGuía para desarrolladores

Timestream

La acción de regla de transmisión de tiempo escribe atributos (medidas) de un mensaje MQTT en una tabla de Amazon Timestream. Para obtener más información acerca de Amazon Timestream, consulte.[¿Qué es Amazon Timestream?](#)

Note

Amazon Timestream no está disponible en todas lasRegión de AWS. Si Amazon Timestream no está disponible en su región, no aparecerá en la lista de acciones de regla.

Los atributos que esta regla almacena en la base de datos Timestream son los que se derivan de la instrucción de consulta de la regla. El valor de cada atributo del resultado de la instrucción de consulta se analiza para inferir su tipo de datos (como en [the section called “DynamoDBv2” \(p. 495\)](#)acción). El valor de cada atributo se escribe en su propio registro en la tabla Timestream. Para especificar o cambiar el tipo de datos de un atributo, utilice la[cast\(\) \(p. 581\)](#)en la instrucción de la consulta. Para obtener más información sobre el contenido de cada registro de transmisión de tiempo, consulte[the section called “Contenido del registro de Amazon Timestream” \(p. 550\).](#)

Note

Con SQL V2 (2016-03-23), valores numéricos que son números enteros, como10.0, se convierten su representación de enteros (10). Lanzarlos explícitamente a unDecimalvalor, por ejemplo, mediante el uso de [laCast\(\) \(p. 581\)](#), no impide este comportamiento; el resultado sigue siendo unIntegerValor . Esto puede provocar errores de discordancia de tipo que impiden que los datos se graben en la base de datos de Timestream. Para procesar de forma fiable los valores numéricos de números completos comoDecimalvalores, utilice SQL V1 (2015-10-08) para la instrucción de consulta de reglas.

Requisitos

Esta acción de regla debe cumplir estos requisitos:

- Un rol de IAM queAWS IoT puede suponer que se lleva a cabo `el timestream:DescribeEndpoints`y `timestream:WriteRecords`soperaciones. Para obtener más información, consulte [Concesión de unAWS IoTrol el acceso que requiere \(p. 473\).](#)

En el navegadorAWS IoTconsola, puede elegir, actualizar o crear un rol para permitirAWS IoTpara llevar a cabo esta acción de regla de.

- Si utilizas un cliente administradoAWS Key Management Service(AWS KMS) para cifrar los datos en reposo de tiempo, el servicio debe tener permiso para poder usar laAWS KMS keyen nombre de la persona que llama. Para obtener más información, consulta[CómoAWSSuso de servicios deAWSKMS](#).

Parámetros

Al crear unAWS IoTcon esta acción, debe especificar la información siguiente:

databaseName

Nombre de una base de datos de Amazon Timestream que contiene la tabla para recibir los registros que crea esta acción. Véase también [tableName](#).

Admite[Plantillas de sustitución \(p. 622\)](#): API yAWS CLIsólo

dimensions

Atributos de metadatos de las series de tiempo que se escriben en cada registro de medida. Por ejemplo, el nombre y la zona de disponibilidad de una instancia EC2 o el nombre del fabricante de un aerogenerador son dimensiones.

name

El nombre de la dimensión de metadatos. Es el nombre de la columna en el registro de tabla de la base de datos.

Las dimensiones no se pueden nombrar:`measure_name`,`measure_value`, o `bientime`. Estos nombres están reservados. Los nombres de las dimensiones no pueden comenzar `ports_` o `measure_value` y no pueden contener los dos puntos (:) personaje.

Admite[Plantillas de sustitución \(p. 622\)](#): No

value

El valor que se va a escribir en esta columna del registro de la base de datos.

Admite[Plantillas de sustitución \(p. 622\)](#): Sí

roleArn

El nombre de recurso de Amazon (ARN) del rol que concedeAWS IoTpermiso para escribir en la tabla de base de datos Timestream. Para obtener más información, consulte [Requisitos \(p. 548\)](#).

Admite[Plantillas de sustitución \(p. 622\)](#): No

tableName

El nombre de la tabla de base de datos en la que se escriben los registros de medidas. Véase también [databaseName](#).

Admite[Plantillas de sustitución \(p. 622\)](#): API yAWS CLIsólo

timestamp

El valor que se va a utilizar para la marca de tiempo de la entrada. Si está en blanco, se utiliza la hora en que se procesó la entrada.

unit

La precisión del valor de marca de tiempo que resulta de la expresión que se describe en `value`.

Valores válidos: `SECONDS` |`MILLISECONDS`|`MICROSECONDS`|`NANOSECONDS`. El valor predeterminado es`MILLISECONDS`.

value

Una expresión que devuelve un valor de tiempo de época larga.

Puede utilizar el[the section called “time_to_epoch \(String, String\)” \(p. 613\)](#)para crear una marca de hora válida a partir de un valor de fecha u hora pasado en la carga útil del mensaje.

Contenido del registro de Amazon Timestream

Los datos escritos en la tabla de Amazon Timestream mediante esta acción incluyen una marca de hora, metadatos de la acción de regla de transmisión de tiempo y el resultado de la instrucción de consulta de la regla.

Para cada atributo (medida) del resultado de la instrucción de consulta, esta acción de regla escribe un registro en la tabla de transmisión de tiempo especificada con estas columnas.

Nombre de la columna	Tipo de atributo	Valor	Comentarios
<i>nombre-dimensión</i>	DIMENSIÓN	El valor especificado en la entrada de acción de regla de transmisión de tiempo.	CadaDimensiónespecificada en la entrada de acción de regla crea una columna en la base de datos Timestream con el nombre de la dimensión.
measure_name	MEASURE_NAME	El nombre del atributo	El nombre del atributo en el resultado de la instrucción de consulta cuyo valor se especifica en el measure_value::datatypecolumn .
measure_value# <i>tipo de datos</i>	MEASURE_VALUE	El valor del atributo del resultado de la instrucción de consulta. El nombre del atributo se encuentra en el measure_namecolumn .	El valor se interpreta* y se convierte en la mejor coincidencia de: <code>bigint</code> , <code>boolean</code> , <code>double</code> , o <code>varchar</code> . Amazon Timestream crea una columna independiente para cada tipo de datos. El valor del mensaje se puede convertir a otro tipo de datos mediante la cast() (p. 581) en la instrucción de consulta de la regla.
tiempo	TIMESTAMP	La fecha y la hora del registro en la base de datos.	Este valor lo asigna el motor de reglas o el timestamppropiedad , si está definida.

* El valor de atributo leído de la carga útil del mensaje se interpreta de la siguiente manera. Consulte[the section called “Ejemplos” \(p. 551\)](#)para obtener una ilustración de cada uno de estos casos.

- Un valor sin cotizar `de true o false` se interpreta como `unboolean`.
- Un numérico decimal se interpreta como `undouble`.
- Un valor numérico sin coma decimal se interpreta como `unbigint`.
- Una cadena citada se interpreta como `unvarchar`.
- Los objetos y los valores de matriz se convierten en cadenas JSON y se almacenan como `varchar`.

Ejemplos

En el siguiente ejemplo de JSON se define una acción de regla de transmisión de tiempo con una plantilla de sustitución en un AWS IoT Rule.

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

El uso de la acción de regla de tema de transmisión de tiempo definida en el ejemplo anterior con la carga útil de mensajes siguiente da como resultado los registros de Amazon Timestream escritos en la tabla siguiente.

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

En la tabla siguiente se muestran las columnas y los registros de la base de datos que utilizan la acción de regla de tema especificada para procesar la carga útil del mensaje anterior que crea. La `device_firmware_sk`y `device_id`columnas son las DIMENSIONES definidas en la acción de regla de tema. La acción de regla de tema de transmisión de tiempo crea el `time`y `measure_name`y `measure_value::*`columnas, que rellena con los valores del resultado de la instrucción de consulta de la acción de regla de tema.

device_firmv	device_id	measure_na	measure_va	measure_va	measure_va	measure_va	tiempo
			bigint	varchar	double	boolean	
Mis metadatos Estáticos	Consola IoT - 159 Ejemplo 738-0	complex_value		{"simple_element": 42, "array_of_integers": [23,36,56,72], "matriz de cadenas": ["rojo", "verde", "azul "]}	-		2020-08-26 22:42:16 .423000000
Mis metadatos Estáticos	Consola IoT - 159 Ejemplo 738-0	integer_value_as_string	123456789012		-		2020-08-26 22:42:16 .423000000
Mis metadatos Estáticos	Consola IoT - 159 Ejemplo 738-0	boolean_value	-	-	TRUE		26-08-2020
Mis metadatos Estáticos	Consola IoT - 159 Ejemplo 738-0	integer_value	123456789012	-	-		26-08-2020
Mis metadatos Estáticos	Consola IoT - 159 Ejemplo 738-0	string_value	-	valor de cadena	-	-	26-08-2020
Mis metadatos Estáticos	Consola IoT - 159 Ejemplo 738-0	array_of_integers	[23,36,56,72]	-	-	-	26-08-2020
Mis metadatos Estáticos	Consola IoT - 159 Ejemplo 738-0	matriz de cadenas	-	["rojo", "verde", "azul"]	-	-	26-08-2020
Mis metadatos Estáticos	Consola IoT - 159 Ejemplo 738-0	boolean_value_as_string	TRUE	-	-	-	26-08-2020
Mis metadatos Estáticos	Consola IoT - 159 Ejemplo 738-0	double_value	-	123,456789012			26-08-2020

device_firmw	device_id	measure_na	measure_va	measure_va	measure_va	measure_val	tiempo
		bigint	varchar	double	boolean		
Mis metadatos Estáticos	Consola IoT - 159 Ejemplo 738-0	double_value_as_string	123.45679	-	-		26-08-2020

Solución de problemas de las reglas

Si tiene algún problema con las reglas, debe habilitar CloudWatch Registros. Puede analizar los registros para determinar si el problema es la autorización o si, por ejemplo, la condición de cláusula WHERE no ha coincidido. Para obtener más información, consulte [Configuración de CloudWatch Logs](#).

Acceso a recursos entre cuentas mediante AWS IoTReglas de

Puede configurar AWS IoTReglas para el acceso entre cuentas, de modo que los datos ingeridos en temas MQTT de una cuenta se puedan enrutar a AWS servicios, como Amazon SQS y Lambda, de otra cuenta. A continuación, se explica cómo configurar AWS IoTReglas para la ingestión de datos entre cuentas, desde un tema MQTT de una cuenta hasta un destino de otra cuenta.

Las reglas multicuenta se pueden configurar mediante [permisos basados en recursos](#) en el recurso de destino. Por lo tanto, solo se pueden habilitar los destinos que admiten permisos basados en recursos para el acceso multicuenta con AWS IoTReglas. Los destinos admitidos incluyen Amazon SQS, Amazon SNS, Amazon S3 y AWS Lambda.

Note

Debe definir la regla en la misma Región de AWS como recurso de otro servicio, de modo que la acción de regla pueda interactuar con ese recurso. Para obtener más información acerca de AWS IoT acciones de reglas de, consulte [AWS IoT Acciones de reglas de \(p. 479\)](#).

Requisitos previos

- Estar familiarizado con [AWS IoTReglas de](#)
- Comprensión de IAM [Usuarios de, Roles de de, y permiso basado en recursos](#)
- Habiendo [AWS CLI](#) instalado

Configuración multicuenta para Amazon SQS

Escenario: La cuenta A envía datos de un mensaje MQTT a la cola de Amazon SQS de la cuenta B.

Cuenta de AWS	Cuenta denominada	Descripción
1111-1111-1111	Cuenta A	Acción de la regla: sqs : SendMessage
2222-2222-2222	Cuenta B	Cola de Amazon SQS

Cuenta de AWS	Cuenta denominada	Descripción
		<ul style="list-style-type: none"> ARN: <code>arn:aws:sqs:region:2222-2222-2222:ExampleQueue</code> URL: <code>https://sqs.region.amazonaws.com/2222-2222-2222/ExampleQueue</code>

Realice las tareas de la cuenta A

Nota

Para ejecutar los siguientes comandos, el usuario de IAM debe tener permisos `paraiot:CreateTopicRule` con el nombre de recurso de Amazon (ARN) de la regla como recurso y permisos para `iam:PassRole` acción con un recurso como ARN del rol.

1. [Configurar AWS CLI](#) utilizando el usuario de IAM de la cuenta A.
2. Crear un rol de IAM en el que confíe AWS IoT adjunta una política que permite el acceso a la cola Amazon SQS de la cuenta B. Consulte comandos y documentos de políticas de ejemplo en [Concesión AWS IoT acceso requerido](#).
3. Para crear una regla que se adjunta a un tema, ejecute la comando `create-topic-rule`.

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

Ejemplo de archivo de carga con una regla que inserta todos los mensajes enviados a la `iot/test` en la cola de Amazon SQS especificada. La instrucción SQL filtra los mensajes, y el ARN del rol concede AWS IoT permisos para agregar el mensaje a la cola de Amazon SQS.

```
{
  "sql": "SELECT * FROM 'iot/test'",
  "ruleDisabled": false,
  "awsIotSqlVersion": "2016-03-23",
  "actions": [
    {
      "sq": {
        "queueUrl": "https://sqs.region.amazonaws.com/2222-2222-2222/ExampleQueue",
        "roleArn": "arn:aws:iam::1111-1111-1111:role/my-iot-role",
        "useBase64": false
      }
    }
  ]
}
```

Para obtener más información sobre cómo definir una acción de Amazon SQS en un AWS IoT regla, véase [AWS IoT acciones de reglas - Amazon SQS](#).

Realice las tareas de la cuenta B

1. [Configurar AWS CLI](#) utilizando el usuario de IAM de la cuenta B.
2. Para conceder permisos para el recurso de cola de Amazon SQS a la cuenta A, ejecute la comando `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
```

Configuración entre cuentas para Amazon SNS

Escenario: La cuenta A envía datos de un mensaje MQTT a un tema de Amazon SNS de la cuenta B.

Cuenta de AWS	Cuenta denominada	Descripción
1111-1111-1111	Cuenta A	Acción de la regla: sns:Publish
2222-2222-2222	Cuenta B	ARN de tema de Amazon SNS: <i>arn:aws:sns:region:2222-2222-2222:ExampleTopic</i>

Realice las tareas de la cuenta A

Notas

Para ejecutar los siguientes comandos, el usuario de IAM debe tener permisos para iot:CreateTopicRule con ARN de regla como recurso y permisos para iam:PassRole acción con un recurso como ARN de rol.

1. [Configurar AWS CLI](#) utilizando el usuario de IAM de la cuenta A.
2. Crear un rol de IAM en el que confíe AWS IoT adjunta una política que permite acceder al tema Amazon SNS de la cuenta B. Por ejemplo, comandos y documentos de políticas, consulte [Concesión AWS IoT acceso requerido](#).
3. Para crear una regla que se adjunta a un tema, ejecute la comando `create-topic-rule`.

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

Ejemplo de archivo de carga con una regla que inserta todos los mensajes enviados a la `iot/test` del tema de Amazon SNS especificado. La instrucción SQL filtra los mensajes, y el ARN del rol concede a permiso para AWS IoT permisos para enviar el mensaje al tema de 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 obtener más información sobre cómo definir una acción de Amazon SNS en un AWS IoT regla, véase [AWS IoT acciones de reglas - Amazon SNS](#).

Realice las tareas de la cuenta B

1. [Configurar AWS CLI](#) utilizando el usuario de IAM de la cuenta B.
2. Para conceder permiso en el recurso de tema de Amazon SNS a la cuenta A, ejecute el comando `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
```

Configuración entre cuentas para Amazon S3

Escenario: La cuenta A envía datos de un mensaje MQTT a un bucket de Amazon S3 de la cuenta B.

Cuenta de AWS	Cuenta denominada	Descripción
1111-1111-1111	Cuenta A	Acción de la regla:s3:PutObject
2222-2222-2222	Cuenta B	ARN del bucket de Amazon S3:arn:aws:s3:::ExampleBucket

Realice las tareas de la cuenta A

Nota

Para ejecutar los siguientes comandos, el usuario de IAM debe tener permisos paraiot:CreateTopicRule con la regla ARN como recurso y permisos paraiam:PassRole acción con un recurso como ARN de rol.

1. [Configurar AWS CLI](#) utilizando el usuario de IAM de la cuenta A.
2. Crear un rol de IAM en el que confíe AWS IoT adjunta una política que permite el acceso al bucket de Amazon S3 de la cuenta B. Por ejemplo, comandos y documentos de políticas, consulte [Concesión AWS IoT acceso requerido](#).
3. Para crear una regla asociada al bucket de S3 de destino, ejecute la comando [create-topic-rule](#).

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

Ejemplo de archivo de carga con una regla que inserta todos los mensajes enviados a la IoT/test en el bucket de Amazon S3 especificado. La instrucción SQL filtra los mensajes, y el ARN del rol concede a permiso para AWS IoT permisos para agregar el mensaje al bucket de 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 obtener más información sobre cómo definir una acción de Amazon S3 en un AWS IoT regla, véase [AWS IoT Acciones de regla: Amazon S3](#).

Realice las tareas de la cuenta B

1. [ConfigurarAWS CLI](#)utilizando el usuario de IAM de la cuenta B.
2. Cree una política de bucket que confíe en el principal de la cuenta A.

Ejemplo de archivo de carga que define una política de bucket que confía en el principal de otra cuenta.

```
{  
    "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 obtener más información, consulte[Ejemplos de política de bucket](#).

3. Para adjuntar la política de bucket al bucket especificado, ejecute la[comando put-bucket-policy](#).

```
aws s3api put-bucket-policy --bucket ExampleBucket --policy file://./my-bucket-  
policy.json
```

4. Para que el acceso multicuenta funcione, asegúrate de tener el correctoBlock all public access (Bloquear todo el acceso público)Configuración del . Para obtener más información, consulte[Prácticas recomendadas de seguridad para Amazon S3](#).

Configuración entre cuentas paraAWS Lambda

Escenario: La cuenta A invoca unAWS Lambdafunción de la cuenta B, transmitiendo un mensaje MQTT.

Cuenta de AWS	Cuenta denominada	Descripción
1111-1111-1111	Cuenta A	Acción de la regla:lambda:InvokeFunction
2222-2222-2222	Cuenta B	ARN de la función Lambda: <i>arn:aws:lambda:region:2222-2222-2222:function:example function</i>

Realice las tareas de la cuenta A

Notas

Para ejecutar los siguientes comandos, el usuario de IAM debe tener permisos para[iot:CreateTopicRule](#)con ARN de regla como recurso y permisos para[iam:PassRole](#)acción con recurso como ARN de rol.

1. [ConfigurarAWS CLI](#)utilizando el usuario de IAM de la cuenta A.

- Ejecute la comando `create-topic-rule`para crear una regla que defina el acceso entre cuentas a la función Lambda de la cuenta B.

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

Ejemplo de archivo de carga con una regla que inserta todos los mensajes enviados a `aiot/test` en la función Lambda especificada. La instrucción SQL filtra los mensajes, y el ARN del rol concede AWS IoT permiso para pasar los datos a la función 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 obtener más información acerca de cómo definir unAWS Lambda acción en unAWS IoT regla, leer [AWS IoT acciones de Lambda](#).

Realice las tareas de la cuenta B

- ConfigurarAWS CLIutilizando el usuario de IAM de la cuenta B.
- Ejecución deComando add-permission de Lambda darAWS IoT permiso de reglas para activar la función Lambda. Para ejecutar el siguiente comando, el usuario de IAM debe tener permiso para `lambda:AddPermission` action.

```
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"
```

Opciones:

--principal

Este campo da permiso aAWS IoT(representado por `iot.amazonaws.com`) para llamar a la función Lambda.

--source-arn

Este campo garantiza que solo `arn:aws:iot:region:1111-1111-1111:rule/example-rule` enAWS IoT activa esta función Lambda y ninguna otra regla en la misma cuenta o en otra cuenta puede activar esta función Lambda.

--source-account

Este campo garantiza queAWS IoT activa esta función Lambda solo en nombre del `1111-1111-1111` account.

Notas

Si aparece un mensaje de error «No se ha encontrado la regla» de su AWS Lambda consola de función en Configuración, ignore el mensaje de error y proceda a probar la conexión.

Control de errores (acción de error)

Cuando AWS IoT recibe un mensaje de un dispositivo, el motor de reglas realiza comprobaciones para ver si el mensaje coincide con una regla. En caso afirmativo, se evalúa la instrucción de consulta de la regla y se activan las acciones de la regla y se pasa el resultado de la instrucción de consulta.

Si se produce un problema al activar una acción, el motor de reglas activa una acción de error si se especificó una para la regla. Esto podría ocurrir cuando:

- Una regla no dispone de permiso para acceder al bucket de Amazon S3.
- Un error de usuario provoca que se supere el rendimiento aprovisionado de DynamoDB.

Formato de mensaje de acción de error

Se genera un único mensaje por regla y mensaje. Por ejemplo, si se produce un error en dos acciones de regla en la misma regla, la acción de error recibe un mensaje con los dos errores.

El mensaje de acción de error es similar al siguiente ejemplo.

```
{  
    "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: yMah1cwPhqTH267QLPhTKeVPKJB8B05ndBHzOmWtxLTM6uAvwYYuqieAKyb6qRPTxP1tHXCoR4Y=). Message arrived on: error/action, Action: s3, Bucket: us-east-1-s3-verify-user, Key: \"aaa\\\". Value of x-amz-id-2: yMah1cwPhqTH267QLPhTKeVPKJB8B05ndBHzOmWtxLTM6uAvwYYuqieAKyb6qRPTxP1tHXCoR4Y=\"  
        }  
    ]  
}
```

ruleName

El nombre de la regla que activó la acción de error.

tema

El tema en el que se recibió el mensaje original.

cloudwatchTraceld

Una identidad única relacionada con los registros de errores en CloudWatch.

clientId

El ID de cliente del publicador de mensajes.

base64OriginalPayload

La carga del mensaje original codificada en Base64.

failures

failedAction

El nombre de la acción que no se pudo completar (por ejemplo "S3Action").

failedResource

El nombre del recurso (por ejemplo el nombre de un bucket de S3).

errorMessage

La descripción y explicación del error.

Ejemplo de acción de error

A continuación se muestra un ejemplo de una regla con una acción de error añadida. La siguiente regla tiene una acción que escribe datos de mensajes en una tabla de DynamoDB y una acción de error que escribe datos en un bucket de 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()}"  
            }  
        }  
    ]  
}
```

Puede utilizar cualquier función o sustitución en una instrucción SQL de la acción de error, excepto en funciones externas (por ejemplo, `get_thing_shadow`, `aws_lambda` y `machinelearning_predict`).

Para obtener más información acerca de reglas y cómo especificar una acción de error, consulte [Creación de una regla de AWS IoT](#).

Para obtener más información acerca del uso de CloudWatch para supervisar el éxito o el fracaso de las reglas, consulte [Métricas y dimensiones de AWS IoT \(p. 437\)](#).

Reducción de costos de mensajería con Basic Ingest

Con Basic Ingest, puede enviar datos de dispositivos de forma segura a laAWSservicios admitidos por [Acciones de reglas de AWS IoT \(p. 479\)](#), sin incurrir [costos de mensajería](#). Basic Ingest optimiza el

flujo de datos eliminando el agente de mensajes de publicación/suscripción de la ruta de adquisición, lo que lo hace más rentable.

Utiliza Basic Ingest para enviar mensajes desde tus dispositivos o aplicaciones. Los mensajes tienen nombres de temas que empiezan por \$aws/rules/*rule_name* para sus tres primeros niveles, donde *rule_name* es el nombre de la AWS IoT Regla que desea invocar.

Puede utilizar una regla existente con Basic Ingest añadiendo el prefijo de Basic Ingest (\$aws/rules/*rule_name*) al tema del mensaje que normalmente utilizaría para invocar la regla. Por ejemplo, si tiene una regla llamada BuildingManager que se invoca mediante mensajes con temas como Buildings/Building5/Floor2/Room201/Lights("sql": "SELECT * FROM 'Buildings/#'"), puede invocar la misma regla con Basic Ingest enviando un mensaje con el tema \$aws/rules/BuildingManager/Buildings/Building5/Floor2/Room201/Lights.

Nota:

- Sus dispositivos y sus reglas no pueden suscribirse a temas reservados para Basic Ingest. Para obtener más información, consulte [Temas reservados \(p. 100\)](#).
- Si necesita un agente de publicación/suscripción para distribuir mensajes a varios suscriptores (por ejemplo para entregar mensajes a otros dispositivos y al motor de reglas), debe seguir utilizando el agente de mensajes de AWS IoT para gestionar la distribución de los mensajes. Sin embargo, asegúrese de publicar sus mensajes en temas que no sean temas de Basic Ingest.

Uso de Basic Ingest

Antes de utilizar Basic Ingest, asegúrese de que su dispositivo o su aplicación utiliza una [Política de de \(p. 333\)](#) que tiene permisos de publicación en \$aws/rules/*. De forma alternativa, puede especificar permiso para reglas individuales con \$aws/rules/*rule_name*/* en la política. De lo contrario, los dispositivos y las aplicaciones podrán seguir utilizando las conexiones existentes con AWS IoT Core.

Cuando el mensaje llega al motor de reglas, no existe diferencia alguna en la ejecución o en la gestión de los errores entre reglas invocadas desde Basic Ingest y las invocadas a través de suscripciones al agente de mensajes.

Puede crear reglas para usarlas con Basic Ingest. Tenga en cuenta lo siguiente:

- El prefijo inicial de un tema de Basic Ingest (\$aws/rules/*rule_name*) no está disponible a la función [topic\(Decimal\) \(p. 613\)](#).
- Si define una regla que se invoca solo con Basic Ingest, la cláusula FROM es opcional en la definición de la regla. Seguirá siendo necesaria si la regla también se invoca a través de otros mensajes que deben enviarse por medio del agente de mensajes (por ejemplo, porque esos otros mensajes deben distribuirse a varios suscriptores). Para obtener más información, consulte [Referencia de la SQL de AWS IoT \(p. 562\)](#).
- Los primeros tres niveles del tema de Basic Ingest (\$aws/rules/*rule_name*) no se cuentan en cuanto al límite de longitud de 8 segmentos o en cuanto al límite de caracteres total de 256 para un tema. De lo contrario, se aplican las mismas restricciones que se documentan en [AWS IoT Límites](#).
- Si se recibe un mensaje con un tema de Basic Ingest que especifica una regla no activa o una regla que no existe, se crea un registro de errores en un registro de Amazon CloudWatch que le ayudará con la tarea de depuración. Para obtener más información, consulte [Entradas del registro del motor de reglas \(p. 457\)](#). Se muestra una métrica RuleNotFound, en la cual podrá crear alarmas. Para obtener más información, consulte [Métricas de la regla \(p. 438\)](#).
- Seguirá pudiendo publicar con QoS 1 en temas de Basic Ingest. Recibirá PUBACK después de que el mensaje se entregue correctamente al motor de reglas. Recibir PUBACK no significa que las acciones relacionadas con la regla se hayan realizado correctamente. Puede configurar una acción de error para gestionar los errores cuando se ejecuta una acción. Para obtener información, consulte [Control de errores \(acción de error\) \(p. 559\)](#).

Referencia de la SQL de AWS IoT

En AWS IoT, las reglas se definen utilizando una sintaxis similar a SQL. Las instrucciones SQL se componen de tres tipos de cláusulas:

SELECT

Obligatorio. Extrae información de la carga útil de un mensaje entrante y realiza transformaciones de la información. Los mensajes que se van a utilizar se identifican mediante el [Filtro de temas \(p. 99\)](#) especificado en la cláusula FROM.

La cláusula SELECT admite [Tipos de datos \(p. 565\)](#), [Operadores \(p. 569\)](#), [Funciones \(p. 575\)](#), [Literales \(p. 619\)](#), [Instrucciones case \(p. 619\)](#), [Extensiones JSON \(p. 620\)](#), [Plantillas de sustitución \(p. 622\)](#), [Consultas de objetos anidados \(p. 623\)](#), y [Cargas binarias \(p. 624\)](#).

FROM

El mensaje MQTT [Filtro de temas \(p. 99\)](#) que identifica los mensajes de los que extraer datos. La regla se activa para cada mensaje enviado a un tema de MQTT que coincide con el filtro de temas especificado aquí. Obligatorio para reglas que se activan mediante mensajes que pasan por el agente de mensajes. Opcional para reglas que solo se activan mediante la característica [Basic Ingest \(p. 560\)](#).

WHERE

(Opcional) Agrega lógica condicional que determina si se llevan a cabo las acciones especificadas por una regla.

La cláusula WHERE admite [Tipos de datos \(p. 565\)](#), [Operadores \(p. 569\)](#), [Funciones \(p. 575\)](#), [Literales \(p. 619\)](#), [Instrucciones case \(p. 619\)](#), [Extensiones JSON \(p. 620\)](#), [Plantillas de sustitución \(p. 622\)](#) y [Consultas de objetos anidados \(p. 623\)](#).

Un ejemplo de instrucción SQL tiene este aspecto:

```
SELECT color AS rgb FROM 'topic/subtopic' WHERE temperature > 50
```

Un mensaje MQTT de ejemplo (también denominado carga de entrada) tiene este aspecto:

```
{  
    "color": "red",  
    "temperature": 100  
}
```

Si este mensaje se publica en el tema 'topic/subtopic', la regla se activa y se evalúa la instrucción SQL. La instrucción SQL extrae el valor de la propiedad color si la propiedad "temperature" es superior a 50. La cláusula WHERE especifica la condición temperature > 50. La palabra clave AS cambia el nombre de la propiedad "color" a "rgb". El resultado (también denominado carga de salida) tiene este aspecto:

```
{  
    "rgb": "red"  
}
```

Estos datos se reenvían después a la acción de la regla, que envía los datos para seguirlos procesando. Para obtener más información sobre las acciones de las reglas, consulte [Acciones de reglas de AWS IoT \(p. 479\)](#).

Note

Actualmente, los comentarios no se admiten en AWS IoT SQL.

Los nombres de atributos con espacios en ellos no se pueden utilizar como nombres de campo en la instrucción SQL. Aunque la carga útil entrante puede tener nombres de atributos con espacios en ellos, estos nombres no se pueden utilizar en la instrucción SQL. Sin embargo, se pasarán a la carga útil saliente si se utiliza una especificación de nombre de campo comodín (*).

Cláusula SELECT

La cláusula SELECT de AWS IoT es básicamente la misma que la cláusula SELECT ANSI SQL con algunas diferencias menores.

La cláusula SELECT admite [Tipos de datos \(p. 565\)](#), [Operadores \(p. 569\)](#), [Funciones \(p. 575\)](#), [Literales \(p. 619\)](#), [Instrucciones case \(p. 619\)](#), [Extensiones JSON \(p. 620\)](#), [Plantillas de sustitución \(p. 622\)](#), [Consultas de objetos anidados \(p. 623\)](#), y [Cargas binarias \(p. 624\)](#).

Puede utilizar la cláusula SELECT para extraer información de los mensajes MQTT de entrada. También puede utilizar SELECT * para recuperar toda la carga del mensaje de entrada. Por ejemplo:

```
Incoming payload published on topic 'topic/subtopic': {"color":"red", "temperature":50}
SQL statement: SELECT * FROM 'topic/subtopic'
Outgoing payload: {"color":"red", "temperature":50}
```

Si la carga es un objeto JSON, puede hacer referencia a claves en el objeto. La carga de salida contiene el par clave-valor. Por ejemplo:

```
Incoming payload published on topic 'topic/subtopic': {"color":"red", "temperature":50}
SQL statement: SELECT color FROM 'topic/subtopic'
Outgoing payload: {"color":"red"}
```

Puede utilizar la palabra clave AS para cambiar el nombre de las claves. Por ejemplo:

```
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"}
```

Puede seleccionar varios elementos separándolos con una coma. Por ejemplo:

```
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}
```

Puede seleccionar varios elementos incluido ** para agregar elementos a la carga de entrada. Por ejemplo:

```
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}
```

Puede utilizar la palabra clave "VALUE" para generar cargas de salida que no sean objetos JSON. Con la versión 2015-10-08 de SQL, solo se puede seleccionar un elemento. Con la versión SQL 2016-03-23 o posterior, también puede seleccionar una matriz para generar como objeto de nivel superior.

Example

```
Incoming payload published on topic 'topic/subtopic': {"color":"red", "temperature":50}
```

```
SQL: SELECT VALUE color FROM 'topic/subtopic'  
Outgoing payload: "red"
```

Puede utilizar la sintaxis '.' para explorar objetos JSON anidados en la carga de entrada. Por ejemplo:

```
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 obtener información sobre cómo utilizar nombres de objetos y propiedades JSON que incluyen caracteres reservados, como números o el guión (menos), consulte [Extensiones JSON \(p. 620\)](#)

Puede utilizar funciones (consulte [Funciones \(p. 575\)](#)) para transformar la carga de entrada. Puede utilizar paréntesis para realizar agrupaciones. Por ejemplo:

```
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

La cláusula FROM suscribe a su regla a un [Tema de \(p. 98\)](#) o [Filtro de temas \(p. 99\)](#). Debe incluir el tema o el filtro de temas entre comillas simples (''). La regla se activa para cada mensaje enviado a un tema de MQTT que coincide con el filtro de temas especificado aquí. Un filtro de temas le permite suscribirse a un grupo de temas similares.

Ejemplo:

Carga de entrada publicada en el tema 'topic/subtopic': {temperature: 50}

Carga de entrada publicada en el tema 'topic/subtopic-2': {temperature: 50}

SQL: "SELECT temperature AS t FROM 'topic/subtopic'".

La regla se suscribe a 'topic/subtopic', por lo que la carga de entrada se pasa a la regla. La carga de salida, que se pasa a las acciones de regla, es: {t: 50}. La regla no está suscrita a 'topic/subtopic-2' por lo que la regla no se activa para el mensaje publicado en 'topic/subtopic-2'.

Ejemplo de comodín #:

Puede utilizar el carácter comodín '#' (multinivel) para buscar coincidencias con uno o varios elementos de ruta en particular:

Carga de entrada publicada en el tema 'topic/subtopic': {temperature: 50}.

Carga de entrada publicada en el tema 'topic/subtopic-2': {temperature: 60}.

Carga de entrada publicada en el tema 'topic/subtopic-3/details': {temperature: 70}.

Carga de entrada publicada en el tema 'topic-2/subtopic-x': {temperature: 80}.

SQL: "SELECT temperature AS t FROM 'topic/#'".

La regla está suscrita a cualquier tema que empiece por 'topic', por lo que se ejecuta tres veces, enviando cargas útiles salientes de {t: 50} (para tema/subtema), {t: 60} (para el tema/subtema 2), y {t: 70} (para el tema/subtema 3/detalles) a sus acciones. No está suscrita a 'topic-2/subtopic-x', por lo que la regla no se activa para el mensaje {temperature: 80}.

Ejemplo de comodín +:

Puede utilizar el carácter comodín '+' (nivel único) para buscar coincidencias con cualquier elemento de ruta en particular:

Carga de entrada publicada en el tema 'topic/subtopic': {temperature: 50}.

Carga de entrada publicada en el tema 'topic/subtopic-2': {temperature: 60}.

Carga de entrada publicada en el tema 'topic/subtopic-3/details': {temperature: 70}.

Carga de entrada publicada en el tema 'topic-2/subtopic-x': {temperature: 80}.

SQL: "SELECT temperature AS t FROM 'topic/+'".

La regla está suscrita a todos los temas que tengan dos elementos de ruta donde el primer elemento sea 'topic'. La regla se ejecuta para los mensajes enviados a 'topic/subtopic' y 'topic/subtopic-2', pero no 'topic/subtopic-3/details' (tiene más niveles que el filtro de temas) o 'topic-2/subtopic-x' (no empieza con topic).

Cláusula WHERE

La cláusula WHERE determina si se llevan a cabo las acciones especificadas por una regla. Si la cláusula WHERE se evalúa en true, se llevan a cabo las acciones de la regla. De lo contrario, las acciones de la regla no se llevan a cabo.

La cláusula WHERE admite [Tipos de datos \(p. 565\)](#), [Operadores \(p. 569\)](#), [Funciones \(p. 575\)](#), [Literales \(p. 619\)](#), [Instrucciones case \(p. 619\)](#), [Extensiones JSON \(p. 620\)](#), [Plantillas de sustitución \(p. 622\)](#) y [Consultas de objetos anidados \(p. 623\)](#).

Ejemplo:

Carga de entrada publicada en topic/subtopic: {"color": "red", "temperature": 40}.

SQL: SELECT color AS my_color FROM 'topic/subtopic' WHERE temperature > 50 AND color <> 'red'.

En este caso, la regla se activa, pero las acciones especificadas por la regla no se llevarán a cabo. No habrá carga de salida.

Puede utilizar funciones y operadores en la cláusula WHERE. Sin embargo, no puede hacer referencia a ningún alias creado con la palabra clave AS en la cláusula SELECT. La cláusula WHERE se evalúa en primer lugar para determinar si SELECT debe evaluarse.

Tipos de datos

El motor de reglas de AWS IoT admite todos los tipos de datos JSON.

Tipos de datos admitidos

Tipo	Significado
Int	Un valor discreto Int. de 34 dígitos como máximo.
Decimal	Un valor Decimal con una precisión de 34 dígitos, con una magnitud no nula mínima de 1E-999 y una magnitud máxima de 9.999... E999.
	<p>Note</p> <p>Algunas funciones devuelven Decimal valores con doble precisión en lugar de una precisión de 34 dígitos.</p>

Tipo	Significado
	Con SQL V2 (2016-03-23), valores numéricos que son números enteros, como <code>10.0</code> , se procesan como <code>unIntValor (10)</code> en lugar de lo esperado <code>DecimalValor (10.0)</code> . Para procesar de forma fiable los valores numéricos de números completos como <code>DecimalValores</code> , utilice SQL V1 (2015-10-08) para la instrucción de consulta de reglas.
<code>Boolean</code>	<code>True</code> o bien <code>False</code> .
<code>String</code>	Una cadena UTF-8.
<code>Array</code>	Una serie de valores que no han de tener obligatoriamente el mismo tipo.
<code>Object</code>	Un valor JSON compuesto de una clave y un valor. Las claves deben ser cadenas. Los valores pueden ser de cualquier tipo.
<code>Null</code>	Valor <code>Null</code> , tal como lo define JSON. Es un valor real que representa la ausencia de valor. El usuario puede crear explícitamente un valor <code>Null</code> especificando la palabra clave <code>Null</code> en la instrucción SQL. Por ejemplo: "SELECT <code>NULL</code> AS <code>n</code> FROM ' <code>topic/subtopic</code> '"
<code>Undefined</code>	No es un valor. No se representa explícitamente en JSON, salvo que se omita el valor. Por ejemplo, en el objeto <code>{"foo": null}</code> , la clave "foo" devuelve <code>NULL</code> , pero la clave "bar" devuelve <code>Undefined</code> . Internamente, el lenguaje SQL trata a <code>Undefined</code> como un valor, pero este no se puede representar en JSON, por lo que cuando se serializa en JSON, los resultados son <code>Undefined</code> . <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <code>{"foo":null, "bar":undefined}</code> </div> se serializa en JSON como: <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <code>{"foo":null}</code> </div> Del mismo modo, <code>Undefined</code> se convierte en una cadena vacía cuando se serializa por sí mismo. Las funciones a las que se llama con argumentos no válidos (por ejemplo, tipos erróneos, número de argumentos erróneo, etc.) devuelven <code>Undefined</code> .

Conversiones

En la tabla siguiente se muestra una lista de los resultados que se producen cuando un valor de un tipo se convierte en otro tipo (cuando se da un valor de tipo incorrecto a una función). Por ejemplo, si a la función

de valor absoluto "abs" (que espera un valor `Int` o `Decimal`) se le da un valor `String`, esta función intentará convertir el valor `String` en un valor `Decimal`, de acuerdo con estas reglas. En este caso, 'abs' ("−5.123") se trata como 'abs(-5.123)'.

Note

No hay ningún intento de conversión en `Array`, `Object`, `Null` o `Undefined`.

En valor decimal

Tipo de argumento	Resultado
<code>Int</code>	Valor <code>Decimal</code> sin separador decimal.
<code>Decimal</code>	El valor de origen.
<code>Boolean</code>	<code>Undefined</code> . (Puede utilizar de forma explícita la función <code>cast</code> para transformar <code>true</code> = 1.0, <code>false</code> = 0.0).
<code>String</code>	El motor SQL intentará analizar la cadena como <code>Decimal</code> . AWS IoT intenta analizar las cadenas que coincidan con la expresión regular: ^-?\d+(.\d+)?((?i)E-?\d+)?\$."0", "-1.2", "5E-12" son ejemplos de cadenas que se convierten automáticamente en valores de tipo <code>Decimal</code> .
<code>Matriz</code>	<code>Undefined</code> .
<code>Objeto</code>	<code>Undefined</code> .
<code>Null</code>	<code>Null</code> .
<code>Sin definir</code>	<code>Undefined</code> .

En valor entero

Tipo de argumento	Resultado
<code>Int</code>	El valor de origen.
<code>Decimal</code>	El valor de origen redondeado al valor <code>Int</code> más cercano.
<code>Boolean</code>	<code>Undefined</code> . (Puede utilizar de forma explícita la función <code>cast</code> para transformar <code>true</code> = 1.0, <code>false</code> = 0.0).
<code>String</code>	El motor SQL intentará analizar la cadena como <code>Decimal</code> . AWS IoT intenta analizar las cadenas que coincidan con la expresión regular: ^-?\d+(.\d+)?((?i)E-?\d+)?\$."0", "-1.2", "5E-12" son ejemplos de cadenas que se convierten automáticamente en valores de tipo <code>Decimal</code> . AWS IoT intenta convertir <code>String</code> en un valor <code>Decimal</code> y, a continuación, eliminar los decimales del valor <code>Decimal</code> para obtener un valor <code>Int</code> .
<code>Matriz</code>	<code>Undefined</code> .

Tipo de argumento	Resultado
Objeto	<code>Undefined</code> .
Null	<code>Null</code> .
Sin definir	<code>Undefined</code> .

En valor booleano

Tipo de argumento	Resultado
<code>Int</code>	<code>Undefined</code> . (Puede utilizar explícitamente elcastFunción para transformar 0 = False, any_nonzero_value = True).
<code>Decimal</code>	<code>Undefined</code> . (Puede utilizar explícitamente la función de conversión para transformar 0 = False, any_nonzero_value = True).
<code>Boolean</code>	El valor original.
<code>String</code>	"true" = True y "false" = False (no distingue entre mayúsculas y minúsculas). Otros valores de string son <code>Undefined</code> .
Matriz	<code>Undefined</code> .
Objeto	<code>Undefined</code> .
Null	<code>Undefined</code> .
Sin definir	<code>Undefined</code> .

En cadena

Tipo de argumento	Resultado
<code>Int</code>	Una representación de cadena del valor <code>Int</code> en notación estándar.
<code>Decimal</code>	Una cadena que representa el valor <code>Decimal</code> , posiblemente en notación científica.
<code>Boolean</code>	"true" o "false". Todos en minúscula.
<code>String</code>	El valor original.
Matriz	El valor <code>Array</code> serializado en formato JSON. La cadena obtenida es una lista separada por comas, entre corchetes. Los valores de tipo <code>String</code> se indican entre comillas. No así los valores de tipo <code>Decimal</code> , <code>Int</code> , <code>Boolean</code> y <code>Null</code> .
Objeto	El objeto serializado al formato JSON. La cadena obtenida es una lista separada por comas de pares clave-valor que comienza y termina con llaves. Los valores de tipo <code>String</code> se indican entre comillas.

Tipo de argumento	Resultado
	No así los valores de tipo Decimal, Int, Boolean y Null.
Null	Undefined.
Sin definir	Sin definir.

Operadores

Los operadores siguientes se pueden utilizar en las cláusulas SELECT y WHERE.

Operador AND

Devuelve un resultado Boolean. Realiza una operación AND lógica. Devuelve el valor true si los operandos izquierdo y derecho son true. De lo contrario, devuelve el valor false. Se necesitan operandos de tipo Boolean u operandos de cadena "true" o "false" que no distingan entre mayúsculas y minúsculas.

Sintaxis: *expression AND expression*.

Operador AND

Operando izquierdo	Operando derecho	Salida
Boolean	Boolean	Boolean. True si ambos operandos son true. De lo contrario, devuelve false.
String/Boolean	String/Boolean	Si todas las cadenas son "true" o "false" (no se distingue entre mayúsculas y minúsculas), se convierten en valores de tipo Boolean y se procesan normalmente como <i>boolean AND boolean</i> .
Otro valor	Otro valor	Undefined.

Operador OR

Devuelve un resultado Boolean. Realiza una operación OR lógica. Devuelve el valor true si el operando izquierdo o el operando derecho es true. De lo contrario, devuelve el valor false. Se necesitan operandos de tipo Boolean u operandos de cadena "true" o "false" que no distingan entre mayúsculas y minúsculas.

Sintaxis: *expression OR expression*.

Operador OR

Operando izquierdo	Operando derecho	Salida
Boolean	Boolean	Boolean. True si uno de los operandos es true. De lo contrario, devuelve false.
String/Boolean	String/Boolean	Si todas las cadenas son "true" o "false" (no se distingue entre mayúsculas y minúsculas), se convierten en valores booleanos y se procesan normalmente como <i>boolean OR boolean</i> .

Operando izquierdo	Operando derecho	Salida
Otro valor	Otro valor	<code>Undefined.</code>

Operador NOT

Devuelve un resultado Boolean. Realiza una operación NOT lógica. Devuelve true si el operando es false. De lo contrario, devuelve true. Se necesita un operando Boolean o un operando de cadena "true" o "false" que no distinga entre mayúsculas y minúsculas.

Sintaxis: `NOT expression.`

Operador NOT

Operando	Salida
Boolean	Boolean. True si el operando es false. De lo contrario, devuelve true.
String	Si la cadena es "true" o "false" (no distingue entre mayúsculas y minúsculas), se convierte en el valor booleano correspondiente y se devuelve el valor opuesto.
Otro valor	<code>Undefined.</code>

Operador >

Devuelve un resultado Boolean. Devuelve el valor true si el operando izquierdo es superior al operando derecho. Los dos operandos se convierten en un valor Decimal y, a continuación, se comparan.

Sintaxis: `expression > expression.`

Operador >

Operando izquierdo	Operando derecho	Salida
Int/Decimal	Int/Decimal	Boolean. Devuelve el valor true si el operando izquierdo es superior al operando derecho. De lo contrario, devuelve false.
String/Int/ Decimal	String/Int/ Decimal	Si todas las cadenas se pueden convertir en un valor Decimal y, a continuación, en un valor Boolean. Devuelve el valor true si el operando izquierdo es superior al operando derecho. De lo contrario, devuelve false.
Otro valor	<code>Undefined.</code>	<code>Undefined.</code>

Operador >=

Devuelve un resultado Boolean. Devuelve el valor true si el operando izquierdo es superior o igual al operando derecho. Los dos operandos se convierten en un valor Decimal y, a continuación, se comparan.

Sintaxis: `expression >= expression.`

Operador >=

Operando izquierdo	Operando derecho	Salida
Int/Decimal	Int/Decimal	Boolean. Devuelve el valor true si el operando izquierdo es superior o igual al operando derecho. De lo contrario, devuelve false.
String/Int/ Decimal	String/Int/ Decimal	Si todas las cadenas se pueden convertir en un valor Decimal y, a continuación, en un valor Boolean. Devuelve el valor true si el operando izquierdo es superior o igual al operando derecho. De lo contrario, devuelve false.
Otro valor	Undefined.	Undefined.

Operador <

Devuelve un resultado Boolean. Devuelve el valor true si el operando izquierdo es inferior al operando derecho. Los dos operandos se convierten en un valor Decimal y, a continuación, se comparan.

Sintaxis: *expression < expression*.

Operador <

Operando izquierdo	Operando derecho	Salida
Int/Decimal	Int/Decimal	Boolean. Devuelve el valor true si el operando izquierdo es inferior al operando derecho. De lo contrario, devuelve false.
String/Int/ Decimal	String/Int/ Decimal	Si todas las cadenas se pueden convertir en un valor Decimal y, a continuación, en un valor Boolean. Devuelve el valor true si el operando izquierdo es inferior al operando derecho. De lo contrario, devuelve false.
Otro valor	Undefined	Undefined

Operador <=

Devuelve un resultado Boolean. Devuelve el valor true si el operando izquierdo es inferior o igual al operando derecho. Los dos operandos se convierten en un valor Decimal y, a continuación, se comparan.

Sintaxis: *expression <= expression*.

Operador <=

Operando izquierdo	Operando derecho	Salida
Int/Decimal	Int/Decimal	Boolean. Devuelve el valor true si el operando izquierdo es inferior o igual al operando derecho. De lo contrario, devuelve false.
String/Int/ Decimal	String/Int/ Decimal	Si todas las cadenas se pueden convertir en un valor Decimal y, a continuación, en un valor Boolean. Devuelve

Operando izquierdo	Operando derecho	Salida
		el valor true si el operando izquierdo es inferior o igual al operando derecho. De lo contrario, devuelve false.
Otro valor	<code>Undefined</code>	<code>Undefined</code>

Operador `<>`

Devuelve un resultado Boolean. Devuelve el valor true si los operandos izquierdo y derecho no son iguales. De lo contrario, devuelve el valor false.

Sintaxis: `expression <> expression`.

Operador `<>`

Operando izquierdo	Operando derecho	Salida
<code>Int</code>	<code>Int</code>	True si el operando izquierdo no es igual al operando derecho. De lo contrario, devuelve false.
<code>Decimal</code>	<code>Decimal</code>	True si el operando izquierdo no es igual al operando derecho. De lo contrario, devuelve false. <code>Int</code> se convierte en un valor <code>Decimal</code> antes de la comparación.
<code>String</code>	<code>String</code>	True si el operando izquierdo no es igual al operando derecho. De lo contrario, devuelve false.
<code>Matriz</code>	<code>Matriz</code>	True si los elementos de cada operando no son iguales y no están en el mismo orden. De lo contrario, devuelve false.
<code>Objeto</code>	<code>Objeto</code>	True si las claves y los valores de cada operando no son iguales. De lo contrario, devuelve false. El orden de las claves y los valores no tiene importancia.
<code>Null</code>	<code>Null</code>	False.
Cualquier valor	<code>Undefined</code>	Sin definir.
<code>Undefined</code>	Cualquier valor	Sin definir.
Tipo no coincidente	Tipo no coincidente	True.

Operador `=`

Devuelve un resultado Boolean. Devuelve el valor true si los operandos izquierdo y derecho son iguales. De lo contrario, devuelve el valor false.

Sintaxis: `expression = expression`.

Operador =

Operando izquierdo	Operando derecho	Salida
Int	Int	True si el operando izquierdo es igual al operando derecho. De lo contrario, devuelve false.
Decimal	Decimal	True si el operando izquierdo es igual al operando derecho. De lo contrario, devuelve false. Int se convierte en un valor Decimal antes de la comparación.
String	String	True si el operando izquierdo es igual al operando derecho. De lo contrario, devuelve false.
Matriz	Matriz	True si los elementos de cada operando son iguales y están en el mismo orden. De lo contrario, devuelve false.
Objeto	Objeto	True si las claves y los valores de cada operando son iguales. De lo contrario, devuelve false. El orden de las claves y los valores no tiene importancia.
Cualquier valor	Undefined	Undefined.
Undefined	Cualquier valor	Undefined.
Tipo no coincidente	Tipo no coincidente	False.

Operador +

El símbolo "+" es un operador sobrecargado. Se puede utilizar para la concatenación o la adición de cadenas.

Sintaxis: *expression* + *expression*.

Operador +

Operando izquierdo	Operando derecho	Salida
String	Cualquier valor	Convierte el operando derecho en una cadena que concatena al final del operando izquierdo.
Cualquier valor	String	Convierte el operando izquierdo en una cadena y concatena el operando derecho al final del operando izquierdo convertido.
Int	Int	Valor Int. Agrega ambos operandos.
Int/Decimal	Int/Decimal	Valor Decimal. Agrega ambos operandos.
Otro valor	Otro valor	Undefined.

Operador -

Resta el operando derecho del operando izquierdo.

Sintaxis: *expression* - *expression*.

Operador -

Operando izquierdo	Operando derecho	Salida
Int	Int	Valor Int. Resta el operando derecho del operando izquierdo.
Int/Decimal	Int/Decimal	Valor Decimal. Resta el operando derecho del operando izquierdo.
String/Int/ Decimal	String/Int/ Decimal	Si todas las cadenas se convierten en decimales correctamente, se devuelve un valor Decimal. Resta el operando derecho del operando izquierdo. De lo contrario, devuelve Undefined.
Otro valor	Otro valor	Undefined.
Otro valor	Otro valor	Undefined.

Operador *

Multiplica el operando izquierdo por el operando derecho.

Sintaxis: *expression* * *expression*.

Operador *

Operando izquierdo	Operando derecho	Salida
Int	Int	Valor Int. Multiplica el operando izquierdo por el operando derecho.
Int/Decimal	Int/Decimal	Valor Decimal. Multiplica el operando izquierdo por el operando derecho.
String/Int/ Decimal	String/Int/ Decimal	Si todas las cadenas se convierten en decimales correctamente, se devuelve un valor Decimal. Multiplica el operando izquierdo por el operando derecho. De lo contrario, devuelve Undefined.
Otro valor	Otro valor	Undefined.

Operador /

Divide el operando izquierdo por el operando derecho.

Sintaxis: *expression* / *expression*.

Operador /

Operando izquierdo	Operando derecho	Salida
Int	Int	Valor Int. Divide el operando izquierdo por el operando derecho.
Int/Decimal	Int/Decimal	Valor Decimal. Divide el operando izquierdo por el operando derecho.
String/Int/ Decimal	String/Int/ Decimal	Si todas las cadenas se convierten en decimales correctamente, se devuelve un valor Decimal. Divide el operando izquierdo por el operando derecho. De lo contrario, devuelve Undefined.
Otro valor	Otro valor	Undefined.

Operador %

Devuelve el resto de la división del operando izquierdo por el operando derecho.

Sintaxis: `expression % expression`.

Operador %

Operando izquierdo	Operando derecho	Salida
Int	Int	Valor Int. Devuelve el resto de la división del operando izquierdo por el operando derecho.
String/Int/ Decimal	String/Int/ Decimal	Si todas las cadenas se convierten en decimales correctamente, se devuelve un valor Decimal. Devuelve el resto de la división del operando izquierdo por el operando derecho. De lo contrario, Undefined.
Otro valor	Otro valor	Undefined.

Funciones

Puede utilizar las funciones integradas siguientes en las cláusulas SELECT o WHERE de sus expresiones SQL.

abs(Decimal)

Devuelve el valor absoluto de un número. Es compatible con la versión 2015-10-08 de SQL y versiones posteriores.

Ejemplo: `abs(-5)` devuelve 5.

Tipo de argumento	Resultado
Int	Int, el valor absoluto del argumento.

Tipo de argumento	Resultado
Decimal	Decimal, el valor absoluto del argumento.
Boolean	Undefined.
String	Decimal. El resultado es el valor absoluto del argumento. Si la cadena no se puede convertir, el resultado es Undefined.
Matriz	Undefined.
Objeto	Undefined.
Null	Undefined.
Sin definir	Undefined.

accountid()

Devuelve el ID de la cuenta que posee esta regla como un valor de tipo `String`. Es compatible con la versión 2015-10-08 de SQL y versiones posteriores.

Ejemplo:

```
accountid() = "123456789012"
```

acos(Decimal)

Devuelve el coseno inverso de un número en radianes. Los argumentos `Decimal` se redondean con doble precisión antes de la aplicación de la función. Es compatible con la versión 2015-10-08 de SQL y versiones posteriores.

Ejemplo: `acos(0) = 1.5707963267948966`

Tipo de argumento	Resultado
Int	Decimal (con doble precisión), el coseno inverso del argumento. Se devuelven resultados imaginarios como Undefined.
Decimal	Decimal (con doble precisión), el coseno inverso del argumento. Se devuelven resultados imaginarios como Undefined.
Boolean	Undefined.
String	Decimal, el coseno inverso del argumento. Si la cadena no se puede convertir, el resultado es Undefined. Se devuelven resultados imaginarios como Undefined.
Matriz	Undefined.
Objeto	Undefined.
Null	Undefined.
Sin definir	Undefined.

asin(Decimal)

Devuelve el seno inverso de un número en radianes. Los argumentos `Decimal` se redondean con doble precisión antes de la aplicación de la función. Es compatible con la versión 2015-10-08 de SQL y versiones posteriores.

Ejemplo: `asin(0) = 0.0`

Tipo de argumento	Resultado
<code>Int</code>	<code>Decimal</code> (con doble precisión), el seno inverso del argumento. Se devuelven resultados imaginarios como <code>Undefined</code> .
<code>Decimal</code>	<code>Decimal</code> (con doble precisión), el seno inverso del argumento. Se devuelven resultados imaginarios como <code>Undefined</code> .
<code>Boolean</code>	<code>Undefined</code> .
<code>String</code>	<code>Decimal</code> (con doble precisión), el seno inverso del argumento. Si la cadena no se puede convertir, el resultado es <code>Undefined</code> . Se devuelven resultados imaginarios como <code>Undefined</code> .
<code>Matriz</code>	<code>Undefined</code> .
<code>Objeto</code>	<code>Undefined</code> .
<code>Null</code>	<code>Undefined</code> .
<code>Sin definir</code>	<code>Undefined</code> .

atan(Decimal)

Devuelve la tangente inversa de un número en radianes. Los argumentos `Decimal` se redondean con doble precisión antes de la aplicación de la función. Es compatible con la versión 2015-10-08 de SQL y versiones posteriores.

Ejemplo: `atan(0) = 0.0`

Tipo de argumento	Resultado
<code>Int</code>	<code>Decimal</code> (con doble precisión), la tangente inversa del argumento. Se devuelven resultados imaginarios como <code>Undefined</code> .
<code>Decimal</code>	<code>Decimal</code> (con doble precisión), la tangente inversa del argumento. Se devuelven resultados imaginarios como <code>Undefined</code> .
<code>Boolean</code>	<code>Undefined</code> .
<code>String</code>	<code>Decimal</code> , la tangente inversa del argumento. Si la cadena no se puede convertir, el resultado es <code>Undefined</code> . Se devuelven resultados imaginarios como <code>Undefined</code> .

Tipo de argumento	Resultado
Matriz	<code>Undefined.</code>
Objeto	<code>Undefined.</code>
Null	<code>Undefined.</code>
Sin definir	<code>Undefined.</code>

atan2(Decimal, Decimal)

Devuelve el ángulo, en radianes, entre el eje x positivo y el punto (x, y) definido en los dos argumentos. El ángulo es positivo para los ángulos en sentido contrario a las agujas del reloj (plano medio superior y > 0) y es negativo para los ángulos que siguen el sentido de las agujas del reloj (plano medio inferior y < 0). Los argumentos `Decimal` se redondean con doble precisión antes de aplicar la función. Es compatible con la versión 2015-10-08 de SQL y versiones posteriores.

Ejemplo: `atan2(1, 0) = 1.5707963267948966`

Tipo de argumento	Tipo de argumento	Resultado
<code>Int/Decimal</code>	<code>Int/Decimal</code>	<code>Decimal</code> (con doble punto (x, y) especificado)
<code>Int/Decimal/String</code>	<code>Int/Decimal/String</code>	<code>Decimal</code> , la tangente cadena no se puede convertir
Otro valor	Otro valor	<code>Undefined.</code>

aws_lambda(functionArn, inputJson)

Llama a la función de Lambda especificada que pasa `inputJson` a la función Lambda y devuelve el JSON generado por la función Lambda.

Argumentos

Argumento	Descripción
<code>functionArn</code>	El ARN de la función de Lambda que se va a llamar. La función Lambda debe devolver datos JSON.
<code>inputJson</code>	La entrada de JSON trasladada a la función Lambda. Para pasar consultas y literales de objetos anidados, debe utilizar la versión SQL 2016-03-23.

Debes conceder AWS IoT `lambda:InvokeFunction` permisos para invocar la función de Lambda especificada. En el siguiente ejemplo, se muestra cómo se puede conceder el permiso `lambda:InvokeFunction` utilizando 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"
```

A continuación, se indican los argumentos del comando add-permission:

--function-name

Nombre de la función Lambda. Agrega un nuevo permiso para actualizar la política de recursos de la función.

--region

La Región de AWS de su cuenta.

--principal

La entidad principal que obtiene el permiso. Este valor debería ser `iot.amazonaws.com` para permitir AWS IoT invocar una función Lambda.

--source-arn

El ARN de la regla. Puede utilizar el `get-topic-rule` AWS CLI para obtener el ARN de una regla.

--source-account

La Cuenta de AWS donde está definida la regla.

--statement-id

Un identificador de instrucción único.

--action

La acción Lambda que desea permitir en esta instrucción. Permitir AWS IoT para invocar una función Lambda, especifique `lambda:InvokeFunction`.

Important

Si agrega un permiso para un AWS IoT principal sin proporcionar el `source-account`, cualquier cuenta de AWS que crea una regla con la acción de Lambda puede activar reglas para invocar la función Lambda desde AWS IoT. Para obtener más información, consulte [Modelo de permisos de Lambda](#).

Dada una carga de mensaje JSON como:

```
{  
    "attribute1": 21,  
    "attribute2": "value"  
}
```

La `aws_lambda` se puede utilizar para llamar a la función Lambda de la siguiente manera.

```
SELECT  
aws_lambda("arn:aws:lambda:us-east-1:account_id:function:lambda_function",  
{"payload":attribute1}) as output FROM 'topic-filter'
```

Si desea pasar la carga del mensaje MQTT completa, puede especificar la carga JSON con `*`, como en el siguiente ejemplo.

```
SELECT  
aws_lambda("arn:aws:lambda:us-east-1:account_id:function:lambda_function", *) as output  
FROM 'topic-filter'
```

`payload.inner.element` selecciona datos de un mensaje publicado en el tema 'tema/subtema'.

`some.values` selecciona datos de la salida generada por la función Lambda.

Note

El motor de reglas limita la duración de la ejecución de las funciones de Lambda. Las llamadas a funciones de Lambda desde las reglas deben completarse en 2000 milisegundos.

bitand(Int, Int)

Ejecuta una operación AND bit a bit en las representaciones de bits de los dos argumentos `Int` (convertidos). Es compatible con la versión 2015-10-08 de SQL y versiones posteriores.

Ejemplo: `bitand(13, 5) = 5`

Tipo de argumento	Tipo de argumento	Resultado
<code>Int</code>	<code>Int</code>	<code>Int</code> , una operación AND.
<code>Int/Decimal</code>	<code>Int/Decimal</code>	<code>Int</code> , una operación AND. Todos los números que no se convierten a <code>Int</code> devuelven <code>undefined</code> . Si al valor <code>Int</code> inferior no se le puede convertir a <code>Int</code> , los dos argumentos no se pueden convertir y el resultado es <code>undefined</code> .
<code>Int/Decimal/String</code>	<code>Int/Decimal/String</code>	<code>Int</code> , una operación AND. Todas las cadenas se redondean al valor <code>Int</code> . Si produce un error en la conversión, el resultado es <code>undefined</code> .
Otro valor	Otro valor	<code>undefined</code> .

bitor(Int, Int)

Realiza una operación OR bit a bit de las representaciones de bit de los dos argumentos. Es compatible con la versión 2015-10-08 de SQL y versiones posteriores.

Ejemplo: `bitor(8, 5) = 13`

Tipo de argumento	Tipo de argumento	Resultado
<code>Int</code>	<code>Int</code>	<code>Int</code> , la operación OR.
<code>Int/Decimal</code>	<code>Int/Decimal</code>	<code>Int</code> , la operación OR. Todos los números que no se convierten a <code>Int</code> devuelven <code>undefined</code> . Si al valor <code>Int</code> inferior no se le puede convertir a <code>Int</code> , los dos argumentos no se pueden convertir y el resultado es <code>undefined</code> .
<code>Int/Decimal/String</code>	<code>Int/Decimal/String</code>	<code>Int</code> , la operación OR. Todas las cadenas se redondean al valor <code>Int</code> . Si produce un error en la conversión, el resultado es <code>undefined</code> .
Otro valor	Otro valor	<code>undefined</code> .

bitxor(Int, Int)

Ejecuta una operación XOR bit a bit en las representaciones de bits de los dos argumentos `Int` (convertidos). Es compatible con la versión 2015-10-08 de SQL y versiones posteriores.

Ejemplo:`bitor(13, 5) = 8`

Tipo de argumento	Tipo de argumento	Resultado
<code>Int</code>	<code>Int</code>	<code>Int</code> , una operación <code>XOR</code> .
<code>Int/Decimal</code>	<code>Int/Decimal</code>	<code>Int</code> , una operación <code>XOR</code> . Los números que no se convierten se redondean al valor <code>Int</code> inferior más cercano.
<code>Int/Decimal/String</code>	<code>Int/Decimal/String</code>	<code>Int</code> , una operación <code>XOR</code> . Las cadenas se convierten en decimales y se redondean al valor <code>Int</code> más cercano. Si se produce un error en la conversión, el resultado es <code>Undefined</code> .
Otro valor	Otro valor	<code>Undefined</code> .

bitnot(Int)

Ejecuta una operación NOT bit a bit en las representaciones de bits del argumento `Int` (convertido). Es compatible con la versión 2015-10-08 de SQL y versiones posteriores.

Ejemplo:`bitnot(13) = 2`

Tipo de argumento	Resultado
<code>Int</code>	<code>Int</code> , una operación NOT bit a bit del argumento.
<code>Decimal</code>	<code>Int</code> , una operación NOT bit a bit del argumento. El valor <code>Decimal</code> se redondea al valor <code>Int</code> inferior más cercano.
<code>String</code>	<code>Int</code> , una operación NOT bit a bit del argumento. Las cadenas se convierten en decimales y se redondean al valor <code>Int</code> inferior más cercano. Si se produce un error en la conversión, el resultado obtenido es <code>Undefined</code> .
Otro valor	Otro valor.

cast()

Convierte un valor de un tipo de datos a otro tipo. Cast se comporta básicamente como las conversiones estándar, salvo que puede convertir números en valores booleanos o viceversa. Si AWS IoT no puede determinar cómo convertir un tipo en otro, el resultado es `Undefined`. Es compatible con la versión 2015-10-08 de SQL y versiones posteriores. Formato: `cast(valor as tipo)`.

Ejemplo:

```
cast(true as Int) = 1
```

Las siguientes palabras clave pueden aparecer después de "as" cuando se llama a cast:

Para las versiones 2015-10-08 y 2016-03-23 de SQL

Palabra clave	Resultado
<code>String</code>	Convierte un valor en <code>String</code> .
<code>Nvarchar</code>	Convierte un valor en <code>String</code> .
<code>Texto</code>	Convierte un valor en <code>String</code> .
<code>Ntext</code>	Convierte un valor en <code>String</code> .
<code>varchar</code>	Convierte un valor en <code>String</code> .
<code>Int</code>	Convierte un valor en <code>Int</code> .
<code>Entero</code>	Convierte un valor en <code>Int</code> .
<code>Double</code>	Convierte un valor en <code>Decimal</code> (con doble precisión).

Además, para SQL versión 2016-03-23

Palabra clave	Resultado
<code>Decimal</code>	Convierte un valor en <code>Decimal</code> .
<code>Bool</code>	Convierte un valor en <code>Boolean</code> .
<code>Boolean</code>	Convierte un valor en <code>Boolean</code> .

Reglas de conversión:

Conversión en decimal

Tipo de argumento	Resultado
<code>Int</code>	Valor <code>Decimal</code> sin separador decimal.
<code>Decimal</code>	El valor de origen. Note Con SQL V2 (2016-03-23), valores numéricos que son números enteros, como <code>10.0</code> , devuelve un <code>Int</code> valor (<code>10</code>) en lugar de lo esperado <code>Decimal</code> valor (<code>10.0</code>). Para convertir de forma fiable valores numéricos de números completos como <code>Decimal</code> valores, utilice SQL V1 (2015-10-08) para la instrucción de consulta de reglas.
<code>Boolean</code>	<code>true = 1.0, false = 0.0</code> .
<code>String</code>	Intenta analizar la cadena como un valor <code>Decimal</code> . AWS IoT intenta analizar las cadenas que coincidan con la expresión regular: <code>^-?d+(\.\d+)?((?i)E-?\d+)?\$. "0", "-1.2", "5E-12"</code> son ejemplos de cadenas que se convierten automáticamente en valores de tipo <code>Decimal</code> .

Tipo de argumento	Resultado
Matriz	<code>Undefined.</code>
Objeto	<code>Undefined.</code>
Null	<code>Undefined.</code>
Sin definir	<code>Undefined.</code>

Conversión en entero

Tipo de argumento	Resultado
<code>Int</code>	El valor de origen.
<code>Decimal</code>	El valor de origen redondeado al valor <code>Int</code> inferior más cercano.
<code>Boolean</code>	<code>true = 1.0, false = 0.0.</code>
<code>String</code>	Intenta analizar la cadena como un valor <code>Decimal</code> . AWS IoT intenta analizar las cadenas que coincidan con la expresión regular: <code>^-?\d+(\.\d+)?((?i)E-?\d+)?\$</code> . "0", "-1.2", "5E-12" son ejemplos de cadenas que se convierten automáticamente en valores de tipo <code>Decimal</code> . AWS IoT intenta convertir la cadena en un valor de tipo <code>Decimal</code> y redondearlo al valor de tipo <code>Int</code> inferior más cercano.
Matriz	<code>Undefined.</code>
Objeto	<code>Undefined.</code>
Null	<code>Undefined.</code>
Sin definir	<code>Undefined.</code>

Conversión a valor `Boolean`

Tipo de argumento	Resultado
<code>Int</code>	<code>0 = False, any_nonzero_value = True.</code>
<code>Decimal</code>	<code>0 = False, any_nonzero_value = True.</code>
<code>Boolean</code>	El valor de origen.
<code>String</code>	"true" = <code>True</code> y "false" = <code>False</code> (no distingue entre mayúsculas y minúsculas). Otros valores de cadena = <code>Undefined.</code>
Matriz	<code>Undefined.</code>
Objeto	<code>Undefined.</code>
Null	<code>Undefined.</code>
Sin definir	<code>Undefined.</code>

Conversión en cadenas

Tipo de argumento	Resultado
<code>Int</code>	Una representación de cadena del valor <code>Int</code> , en notación estándar.
<code>Decimal</code>	Una cadena que representa el valor <code>Decimal</code> , posiblemente en notación científica.
<code>Boolean</code>	" <code>true</code> " o " <code>false</code> ", todo en minúsculas.
<code>String</code>	" <code>true</code> " = <code>True</code> y " <code>false</code> " = <code>False</code> (no distingue entre mayúsculas y minúsculas). Otros valores de cadena = <code>Undefined</code> .
Matriz	La matriz serializada en formato JSON. La cadena obtenida es una lista separada por comas, entre corchetes. Los valores <code>String</code> se indican entre comillas. Los valores <code>Decimal</code> , <code>Int</code> y <code>Boolean</code> no se indican entre comillas.
Objeto	El objeto serializado al formato JSON. La cadena JSON es una lista separada por comas de pares clave-valor que comienza y termina con llaves. Los valores <code>String</code> se indican entre comillas. Los valores <code>Decimal</code> , <code>Int</code> , <code>Boolean</code> y <code>Null</code> no se indican entre comillas.
<code>Null</code>	<code>Undefined</code> .
Sin definir	<code>Undefined</code> .

ceil(`Decimal`)

Redondea el valor `Decimal` indicado al valor `Int` superior más cercano. Es compatible con la versión 2015-10-08 de SQL y versiones posteriores.

Ejemplos:

`ceil(1.2) = 2`

`ceil(-1.2) = -1`

Tipo de argumento	Resultado
<code>Int</code>	<code>Int</code> , el valor del argumento.
<code>Decimal</code>	<code>Int</code> , el valor de <code>Decimal</code> redondeado al valor de tipo <code>Int</code> superior más cercano.
<code>String</code>	<code>Int</code> . La cadena se convierte en <code>Decimal</code> y redondeado al valor superior más cercano <code>Int</code> . Si la cadena no se puede convertir en un valor <code>Decimal</code> , el resultado es <code>Undefined</code> .
Otro valor	<code>Undefined</code> .

chr(String)

Devuelve el carácter ASCII que corresponde al argumento `Int` determinado. Es compatible con la versión 2015-10-08 de SQL y versiones posteriores.

Ejemplos:

```
chr(65) = "A".
```

```
chr(49) = "1".
```

Tipo de argumento	Resultado
<code>Int</code>	El carácter correspondiente al valor ASCII especificado. Si el argumento no es un valor ASCII válido, el resultado es <code>Undefined</code> .
<code>Decimal</code>	El carácter correspondiente al valor ASCII especificado. El argumento <code>Decimal</code> se redondea al valor <code>Int</code> inferior más cercano. Si el argumento no es un valor ASCII válido, el resultado es <code>Undefined</code> .
<code>Boolean</code>	<code>Undefined</code> .
<code>String</code>	Si el valor <code>String</code> puede convertirse en un valor <code>Decimal</code> , se redondea al valor <code>Int</code> inferior más cercano. Si el argumento no es un valor ASCII válido, el resultado es <code>Undefined</code> .
Matriz	<code>Undefined</code> .
Objeto	<code>Undefined</code> .
Null	<code>Undefined</code> .
Otro valor	<code>Undefined</code> .

clientid()

Devuelve el ID del cliente MQTT que envía el mensaje o `n/a` si el mensaje no se ha enviado por MQTT. Es compatible con la versión 2015-10-08 de SQL y versiones posteriores.

Ejemplo:

```
clientid() = "123456789012"
```

concat()

Concatena matrices o cadenas. Esta función acepta cualquier cantidad de argumentos y devuelve un valor `String` o un valor `Array`. Es compatible con la versión 2015-10-08 de SQL y versiones posteriores.

Ejemplos:

```
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	<code>Undefined</code> .
1	El argumento se devuelve sin modificar.
2+	<p>Si alguno de los argumentos es un valor <code>Array</code>, el resultado es una matriz única que contiene todos los argumentos. Si no hay argumentos de tipo <code>Array</code> y al menos un argumento es un valor <code>String</code>, el resultado es la concatenación de las representaciones de <code>String</code> de todos los argumentos. Los argumentos se convierten en cadenas mediante las conversiones estándar indicadas arriba.</p> <p>.</p>

cos(Decimal)

Devuelve el coseno de un número en radianes. Los argumentos `Decimal` se redondean con doble precisión antes de la aplicación de la función. Es compatible con la versión 2015-10-08 de SQL y versiones posteriores.

Ejemplo:

`cos(0) = 1.`

Tipo de argumento	Resultado
<code>Int</code>	<code>Decimal</code> (con doble precisión), el coseno del argumento. Se devuelven resultados imaginarios como <code>Undefined</code> .
<code>Decimal</code>	<code>Decimal</code> (con doble precisión), el coseno del argumento. Se devuelven resultados imaginarios como <code>Undefined</code> .
<code>Boolean</code>	<code>Undefined</code> .
<code>String</code>	<code>Decimal</code> (con doble precisión), el coseno del argumento. Si la cadena no se puede convertir en un valor <code>Decimal</code> , el resultado es <code>Undefined</code> . Se devuelven resultados imaginarios como <code>Undefined</code> .
<code>Matriz</code>	<code>Undefined</code> .
<code>Objeto</code>	<code>Undefined</code> .
<code>Null</code>	<code>Undefined</code> .
<code>Sin definir</code>	<code>Undefined</code> .

cosh(Decimal)

Devuelve el coseno hiperbólico de un número en radianes. Los argumentos `Decimal` se redondean con doble precisión antes de la aplicación de la función. Es compatible con la versión 2015-10-08 de SQL y versiones posteriores.

Ejemplo: `cosh(2.3) = 5.037220649268761.`

Tipo de argumento	Resultado
<code>Int</code>	<code>Decimal</code> (con doble precisión), el coseno hiperbólico del argumento. Se devuelven resultados imaginarios como <code>Undefined</code> .
<code>Decimal</code>	<code>Decimal</code> (con doble precisión), el coseno hiperbólico del argumento. Se devuelven resultados imaginarios como <code>Undefined</code> .
<code>Boolean</code>	<code>Undefined</code> .
<code>String</code>	<code>Decimal</code> (con doble precisión), el coseno hiperbólico del argumento. Si la cadena no se puede convertir en un valor <code>Decimal</code> , el resultado es <code>Undefined</code> . Se devuelven resultados imaginarios como <code>Undefined</code> .
<code>Matriz</code>	<code>Undefined</code> .
<code>Objeto</code>	<code>Undefined</code> .
<code>Null</code>	<code>Undefined</code> .
<code>Sin definir</code>	<code>Undefined</code> .

decode (valor, decodingScheme)

Usar `decode` para decodificar un valor codificado. Si la cadena decodificada es un documento JSON, se devuelve un objeto direccionable. De lo contrario, la cadena decodificada se devuelve como una cadena. La función devuelve `NULL` si la cadena no se puede decodificar.

Es compatible con la versión SQL 2016-03-23 y versiones posteriores.

value

Un valor de cadena o cualquiera de las expresiones válidas, tal y como se define en [Referencia de la SQL de AWS IoT \(p. 562\)](#), que devuelve una cadena.

Esquema de decodificación

Una cadena literal que representa el esquema utilizado para decodificar el valor. En la actualidad, solo se admite '`base64`'.

Ejemplo

En este ejemplo, la carga útil del mensaje incluye un valor codificado.

```
{  
    encoded_temp: "eyAidGVtcGVyYXR1cmUiOiAzMyB9Cg=="}
```

```
}
```

La decode en esta instrucción SQL, decodifica el valor de la carga útil del mensaje.

```
SELECT decode(encoded_temp, "base64").temperature AS temp from 'topic/subtopic'
```

Descodificar el valor de encoded_temp da como resultado el siguiente documento JSON válido, que permite a la instrucción SELECT leer el valor de temperatura.

```
{ "temperature": 33 }
```

El resultado de la instrucción SELECT de este ejemplo se muestra aquí.

```
{ "temp": 33 }
```

Si el valor decodificado no fuera un documento JSON válido, el valor decodificado se devolvería como una cadena.

encode(value, encodingScheme)

Utilice la función encode para codificar la carga, que puede estar constituida por datos que no son JSON, en su representación de cadena basada en el esquema de codificación. Es compatible con la versión SQL 2016-03-23 y versiones posteriores.

value

Cualquiera de las expresiones válidas, tal y como se define en [Referencia de la SQL de AWS IoT \(p. 562\)](#). Puede especificar * para codificar toda la carga, con independencia de si está en formato JSON o no. Si suministra una expresión, el resultado de la evaluación se convierte en una cadena antes de codificarla.

encodingScheme

Una cadena literal que representa el esquema de codificación que desea utilizar. En la actualidad, solo se admite 'base64'.

endswith(String, String)

Devuelve un valor Booleano que indica si el primer argumento String termina con el segundo argumento String. Si alguno de los argumentos es Null o Undefined, el resultado es Undefined. Es compatible con la versión 2015-10-08 de SQL y versiones posteriores.

Ejemplo: endswith("cat", "at") = true.

Tipo de argumento 1	Tipo de argumento 2	Resultado
String	String	True si el primer argumento. De lo contrario, False.
Otro valor	Otro valor	Ambos argumentos son strings. Si ambos cumplen con las reglas de conversión, el resultado es True. Si alguno de los argumentos no cumple con las reglas de conversión, el resultado es False. Si alguno de los argumentos es null o undefined, el resultado es undefined.

exp(Decimal)

Devuelve e elevado al argumento Decimal. Los argumentos Decimal se redondean con doble precisión antes de la aplicación de la función. Es compatible con la versión 2015-10-08 de SQL y versiones posteriores.

Ejemplo: `exp(1) = e.`

Tipo de argumento	Resultado
Int	Decimal (con doble precisión), argumento potencia e.
Decimal	Decimal (con doble precisión), argumento potencia e.
String	Decimal (con doble precisión), argumento potencia e. Si el valor String no se puede convertir en un valor Decimal, el resultado es Undefined.
Otro valor	Undefined.

floor(Decimal)

Redondea a la baja el valor Decimal indicado al valor Int más cercano. Es compatible con la versión 2015-10-08 de SQL y versiones posteriores.

Ejemplos:

`floor(1.2) = 1`

`floor(-1.2) = -2`

Tipo de argumento	Resultado
Int	Int, el valor del argumento.
Decimal	Int, valor Decimal redondeado a la baja al valor Int más próximo.
String	Int. La cadena se convierte en Decimal y redondeado al valor inferior más cercano Int. Si la cadena no se puede convertir en un valor Decimal, el resultado es Undefined.
Otro valor	Undefined.

get

Extrae un valor de un tipo de recopilación (matriz, cadena, objeto). No se aplica ninguna conversión al primer argumento. La conversión se aplica tal y como se documenta en la tabla del segundo argumento. Es compatible con la versión 2015-10-08 de SQL y versiones posteriores.

Ejemplos:

`get(["a", "b", "c"], 1) = "b"`

```
get({ "a": "b" }, "a") = "b"
```

```
get("abc", 1) = «a»
```

Tipo de argumento 1	Tipo de argumento 2	Resultado
Matriz	Cualquier tipo (convertido en <code>Int</code>)	El elemento en el índice proporcionado por el segundo argumento (convertido en <code>Int</code>). Si la conversión es <code>Undefined</code> . Si el índice es negativo, el resultado es <code>Undefined</code> .
Cadena	Cualquier tipo (convertido en <code>Int</code>)	El carácter en el índice proporcionada por el segundo argumento (convertido en <code>Int</code>). Si la conversión obtenida es <code>Undefined</code> , de los límites de la cadena, el resultado es <code>Undefined</code> .
Objeto	<code>String</code> (no se aplica conversión)	El valor almacenado en la clave correspondiente a la clave que se pasa como segundo argumento.
Otro valor	Cualquier valor	<code>Undefined</code> .

get_dynamodb

(`tableName,partitionKeyName,partitionKeyValue,sortKeyName,sortKeyValue,roleArn`)

Recupera datos de una tabla de DynamoDB.`get_dynamodb()` permite consultar una tabla de DynamoDB mientras se evalúa una regla. Puede filtrar o aumentar las cargas útiles de mensajes utilizando datos recuperados de DynamoDB. Es compatible con la versión SQL 2016-03-23 y versiones posteriores.

`get_dynamodb()` utiliza los parámetros siguientes:

`tableName`

Nombre de la tabla de DynamoDB donde efectuar la consulta.

`partitionKeyName`

Nombre de la tabla de particiones. Para obtener más información, consulte [Teclas DynamoDB](#).

`partitionKeyValue`

Valor de la clave de partición utilizada para identificar un registro. Para obtener más información, consulte [Teclas DynamoDB](#).

`sortKeyName`

(Opcional) El nombre de la clave de ordenación. Este parámetro solo es necesario si la tabla de DynamoDB consultada utiliza una clave compuesta. Para obtener más información, consulte [Teclas DynamoDB](#).

`sortKeyValue`

(Opcional) Valor de la clave de ordenación. Este parámetro solo es necesario si la tabla de DynamoDB consultada utiliza una clave compuesta. Para obtener más información, consulte [Teclas DynamoDB](#).

roleArn

El ARN de un rol de IAM que concede acceso a la tabla DynamoDB. El motor de reglas asume este rol para acceder a la tabla DynamoDB en su nombre. Evite usar un rol excesivamente permisivo. Otorgue al rol solo los permisos requeridos por la regla. A continuación se muestra un ejemplo de política que concede acceso a una tabla de DynamoDB.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": "dynamodb:GetItem",  
            "Resource": "arn:aws:dynamodb:aws-region:account-id:table/table-name"  
        }  
    ]  
}
```

Como ejemplo de cómo puede utilizar `get_dynamodb()`, supongamos que tiene una tabla de DynamoDB que contiene el ID de dispositivo e información de ubicación para todos los dispositivos conectados a AWS IoT. La siguiente instrucción SELECT utiliza la función `get_dynamodb()` para recuperar la ubicación del ID de dispositivo especificado:

```
SELECT *, get_dynamodb("InServiceDevices", "deviceId", id,  
"arn:aws:iam::12345678910:role/getdynamo").location AS location FROM 'some/  
topic'
```

Note

- Puede llamar a `get_dynamodb()` un máximo de una vez por instrucción SQL. Llamar a `get_dynamodb()` varias veces en una sola instrucción SQL hace que la regla termine sin invocar ninguna acción.
- Si `get_dynamodb()` devuelve más de 8 KB de datos, no se puede invocar la acción de la regla.

get_secret (secretId, tipo secreto, clave, roleArn)

Recupera el valor del cifrado `SecretString` o `SecretBinary` campo de la versión actual de un secreto en [AWS Secrets Manager](#). Para obtener más información acerca de la creación y el mantenimiento de secretos, consulte [CreateSecret](#), [UpdateSecret](#), y [PutSecretValue](#).

`get_secret()` utiliza los parámetros siguientes:

secretId

Cadena: El nombre de recurso de Amazon (ARN) o el nombre fácil de conocer del secreto que se va a recuperar.

Tipo secreto

Cadena: El tipo de secreto. Valores válidos: `SecretString` | `SecretBinary`.

SecretString

- Para los secretos que crea como objetos JSON mediante las API, el AWS CLI, o el AWS Secrets Manager Consola de :
 - Si especifica un valor para la acción `key`: esta función devuelve el valor de la clave especificada.
 - Si no especifica un valor para la acción `key`, esta función devuelve todo el objeto JSON.

- Para los secretos que crea como objetos que no son JSON mediante las API o elAWS CLI:
 - Si especifica un valor para la acciónkey, esta función falla con una excepción.
 - Si no especifica un valor para la acciónkey, esta función devuelve el contenido del secreto.

SecretBinary

- Si especifica un valor para la acciónkey, esta función falla con una excepción.
- Si no especifica un valor para la acciónkey, esta función devuelve el valor secreto como una cadena UTF-8 codificada en base64.

key

(Opcional) String: El nombre clave dentro de un objeto JSON almacenado en elSecretStringcampo de un secreto. Utilice este valor cuando desee recuperar solo el valor de una clave almacenada en un secreto en lugar de todo el objeto JSON.

Si especifica un valor para este parámetro y el secreto no contiene ningún objeto JSON dentro de suSecretString, esta función falla con una excepción.

roleArn

Cadena: Un ARN de rol

desecretsmanager:GetSecretValueysecretsmanager:DescribeSecretpermisos.

Note

Esta función siempre devuelve la versión actual del secreto (la versión con elAWS CURRENTetiqueta). LaAWS IoTEl motor de reglas almacena en caché cada secreto durante unos 15 minutos como máximo. Como resultado, el motor de reglas puede tardar hasta 15 minutos en actualizar un secreto. Esto significa que si recuperas un secreto hasta 15 minutos después de una actualización conAWS Secrets Manager, esta función podría devolver la versión anterior.

Esta función no está medida y es de uso gratuito, peroAWS Secrets Managerse aplican cargos. Debido al mecanismo de almacenamiento en caché secreto, el motor de reglas llama ocasionalmenteAWS Secrets Manager. Dado que el motor de reglas es un servicio totalmente distribuido, es posible que veas varias llamadas a la API de Secrets Manager desde el motor de reglas durante la ventana de almacenamiento en caché de 15 minutos.

Ejemplos:

Puede utilizar elget_secreten un encabezado de autenticación en una acción de regla HTTPS, como en el siguiente ejemplo de autenticación de clave de API.

```
"API_KEY": "${get_secret('API_KEY', 'SecretString', 'API_KEY_VALUE',  
'arn:aws:iam::12345678910:role/getsecret')}"
```

Para obtener más información acerca de la acción de regla HTTPS, consulte[the section called "HTTP" \(p. 498\)](#).

get_thing_shadow(thingName, shadowName, roleARN)

Devuelve la sombra especificada del objeto especificado. Es compatible con la versión SQL 2016-03-23 y versiones posteriores.

thingName

Cadena: El nombre de la cosa cuya sombra desea recuperar.

shadowName

(Opcional) String: El nombre de la sombra. Este parámetro solo es necesario cuando se hace referencia a sombras con nombre.

roleArn

Cadena: Un ARN de rol deiot:GetThingShadowpermiso.

Ejemplos:

Cuando se utilice con una sombra con nombre, proporcione el parámetro shadowName.

```
SELECT * from 'topic/subtopic'  
WHERE  
    get_thing_shadow("MyThing", "MyThingShadow", "arn:aws:iam::123456789012:role/  
AllowsThingShadowAccess")  
    .state.reported.alarm = 'ON'
```

Cuando se utiliza con una sombra sin nombre, omita el parámetro shadowName.

```
SELECT * from 'topic/subtopic'  
WHERE  
    get_thing_shadow("MyThing", "arn:aws:iam::123456789012:role/AllowsThingShadowAccess")  
    .state.reported.alarm = 'ON'
```

Funciones de hash

AWS IoT ofrece las siguientes funciones de hash:

- md2
- md5
- sha1
- sha224
- sha256
- sha384
- sha512

Todas las funciones de hash esperan un argumento de cadena. El resultado es el valor con hash de dicha cadena. Las conversiones de cadena estándar se aplican a los argumentos que no son cadenas. Todas las funciones de hash son compatibles con la versión 2015-10-08 de SQL y con versiones posteriores.

Ejemplos:

md2("hello") = "a9046c73e00331af68917d3804f70655"

md5("hello") = "5d41402abc4b2a76b9719d911017c592"

indexof(String, String)

Devuelve el primer índice (basado en 0) del segundo argumento como subcadena del primer argumento. Se espera que ambos argumentos sean cadenas. Los argumentos que no sean cadenas están sujetos a las reglas de conversión estándar de cadenas. Esta función no se aplica a matrices, únicamente a cadenas. Es compatible con la versión SQL 2016-03-23 y versiones posteriores.

Ejemplos:

```
indexof("abcd", "bc") = 1
```

isNull()

Devuelve verdadero si el argumento es el valor `Null`. Es compatible con la versión 2015-10-08 de SQL y versiones posteriores.

Ejemplos:

```
isNull(5) = false.
```

```
isNull(null) = true.
```

Tipo de argumento	Resultado
Int	falso
Decimal	falso
Boolean	falso
String	falso
Array	falso
Object	falso
Null	true
Undefined	falso

isUndefined()

Devuelve verdadero si el argumento tiene el valor `undefined`. Es compatible con la versión SQL 2016-03-23 y versiones posteriores.

Ejemplos:

```
isUndefined(5) = false.
```

```
isUndefined(floor([1,2,3])) = true.
```

Tipo de argumento	Resultado
Int	falso
Decimal	falso
Boolean	falso
String	falso
Array	falso
Object	falso

Tipo de argumento	Resultado
Null	false
Undefined	true

length(String)

Devuelve el número de caracteres de la cadena suministrada. Se aplican las reglas de conversión estándar a los argumentos que no sean String. Es compatible con la versión SQL 2016-03-23 y versiones posteriores.

Ejemplos:

```
length("hi") = 2
length(false) = 5
```

ln(Decimal)

Devuelve el logaritmo natural del argumento. Los argumentos Decimal se redondean con doble precisión antes de la aplicación de la función. Es compatible con la versión 2015-10-08 de SQL y versiones posteriores.

Ejemplo: $\ln(e) = 1$.

Tipo de argumento	Resultado
Int	Decimal (con doble precisión), el log natural del argumento.
Decimal	Decimal (con doble precisión), el log natural del argumento.
Boolean	Undefined.
String	Decimal (con doble precisión), el log natural del argumento. Si la cadena no se puede convertir en un valor Decimal, el resultado es Undefined.
Matriz	Undefined.
Objeto	Undefined.
Null	Undefined.
Sin definir	Undefined.

log(Decimal)

Devuelve el logaritmo decimal del argumento. Los argumentos Decimal se redondean con doble precisión antes de la aplicación de la función. Es compatible con la versión 2015-10-08 de SQL y versiones posteriores.

Ejemplo: $\log(100) = 2,0$.

Tipo de argumento	Resultado
<code>Int</code>	<code>Decimal</code> (con doble precisión), el log de base 10 del argumento.
<code>Decimal</code>	<code>Decimal</code> (con doble precisión), el log de base 10 del argumento.
<code>Boolean</code>	<code>Undefined</code> .
<code>String</code>	<code>Decimal</code> (con doble precisión), el log de base 10 del argumento. Si el valor <code>String</code> no se puede convertir en un valor <code>Decimal</code> , el resultado es <code>Undefined</code> .
<code>Matriz</code>	<code>Undefined</code> .
<code>Objeto</code>	<code>Undefined</code> .
<code>Null</code>	<code>Undefined</code> .
<code>Sin definir</code>	<code>Undefined</code> .

lower(`String`)

Muestra la versión en minúsculas del valor `String` indicado. Los argumentos que no son cadenas se convierten en cadenas con las reglas de conversión estándar. Es compatible con la versión 2015-10-08 de SQL y versiones posteriores.

Ejemplos:

```
lower("HELLO") = "hello".
lower(["HELLO"]) = ["hello"].
```

lpad(`String`, `Int`)

Devuelve el argumento `String`, rellenado en el lado izquierdo con el número de espacios especificado por el segundo argumento. El argumento `Int` debe estar comprendido entre 0 y 1000. Si el valor proporcionado se encuentra fuera de este rango válido, el argumento se establece en el valor válido más cercano (0 o 1000). Es compatible con la versión 2015-10-08 de SQL y versiones posteriores.

Ejemplos:

```
lpad("hello", 2) = " hello".
lpad(1, 3) = " 1"
```

Tipo de argumento 1	Tipo de argumento 2	Resultado
<code>String</code>	<code>Int</code>	<code>String</code> , el valor <code>String</code> relleno en el lado izquierdo con un espacio de <code>Int</code> proporcionado.
<code>String</code>	<code>Decimal</code>	El argumento <code>Decimal</code> más cercano y el valor <code>String</code> relleno en el lado izquierdo con el número de espacios de <code>Decimal</code> .

Tipo de argumento 1	Tipo de argumento 2	Resultado
String	String	El segundo argumento que se redondea al valor String se reespecificado en la izquierda. No se puede convertir Undefined.
Otro valor	Int/Decimal/String	El primer valor se convierte mediante las conversiones que se aplica la función LTRIM. Si no se puede convertir, el resultado es Undefined.
Cualquier valor	Otro valor	Undefined.

Itrim(String)

Elimina todos los espacios en blanco del principio (tabuladores y espacios) del valor String proporcionado. Es compatible con la versión 2015-10-08 de SQL y versiones posteriores.

Ejemplo:

`Itrim(" h i ") = "hi".`

Tipo de argumento	Resultado
Int	La representación String de Int con todos los espacios en blanco del principio suprimidos.
Decimal	La representación String de Decimal con todos los espacios en blanco del principio suprimidos.
Boolean	La representación String del valor booleano ("true" o "false") con todos los espacios en blanco del principio suprimidos.
String	El argumento con todos los espacios en blanco del principio suprimidos.
Matriz	La representación String de Array (mediante las reglas de conversión estándar) con todos los espacios en blanco del principio suprimidos.
Objeto	La representación String del objeto (mediante las reglas de conversión estándar) con todos los espacios en blanco del principio suprimidos.
Null	Undefined.
Sin definir	Undefined.

machinelearning_predict (modelId, roleArn, grabar)

Usa `machinelearning_predict` para realizar predicciones con los datos de un mensaje MQTT basado en un modelo de Amazon Machine Learning (Amazon ML). Es compatible con la versión 2015-10-08 de SQL y versiones posteriores. Los argumentos de la función `machinelearning_predict` son:

modelId

El ID del modelo en el que se ejecutará la predicción. El punto de enlace en tiempo real del modelo debe estar activado.

roleArn

El papel de IAM que tiene una política `com:AmazonMachineLearning:Predict` que permite tener permisos `MLModelPermissions` y permite tener acceso al modelo en el que se ejecuta la predicción.

record

Los datos que van a transmitirse a la API de previsión de Amazon ML. Debe representarse como un objeto JSON de capa única. Si el registro es un objeto JSON de varias capas, el registro se aplana serializando sus valores. Por ejemplo, el JSON siguiente:

```
{ "key1": { "innerKey1": "value1"}, "key2": 0}
```

se convertiría en:

```
{ "key1": "{\"innerKey1\": \"value1\"}", "key2": 0}
```

La función devuelve un objeto JSON con los campos siguientes:

predictedLabel

La clasificación de la entrada en función del modelo.

details

Contiene los atributos siguientes:

PredictiveModelTipo

El tipo de modelo. Los valores válidos son REGRESSION, BINARY, MULTICLASS.

Algoritmo

El algoritmo utilizado por Amazon ML para realizar las predicciones. El valor debe ser SGD.
predictedScores

Contiene la puntuación de clasificación bruta correspondiente a cada etiqueta.

predictedValue

Valor que Amazon ML prevé.

mod(Decimal, Decimal)

Devuelve el resto de la división del primer argumento por el segundo argumento. Es igual que [remainder\(Decimal, Decimal\) \(p. 604\)](#). También puede utilizar "%" como un operador infijo para la misma funcionalidad modulo. Es compatible con la versión 2015-10-08 de SQL y versiones posteriores.

Ejemplo: `mod(8, 3) = 2`.

Operando izquierdo	Operando derecho	Salida
Int	Int	Int, el primer y el segundo argumentos deben ser enteros.

Operando izquierdo	Operando derecho	Salida
Int/Decimal	Int/Decimal	Decimal, el primer argumento para los que quiere ejemplos.
String/Int/Decimal	String/Int/Decimal	Si todas las cadenas se pasan a la función modulo se devolverá el resultado. De lo contrario, se devolverá undefined.
Otro valor	Otro valor	Undefined.

nanvl (AnyValue,AnyValue)

Devuelve el primer argumento si es un Decimal válido. De lo contrario, se devuelve el segundo argumento. Es compatible con la versión 2015-10-08 de SQL y versiones posteriores.

Ejemplo: `Nanvl(8, 3) = 8.`

Tipo de argumento 1	Tipo de argumento 2	Salida
Sin definir	Cualquier valor	El segundo argumento.
Null	Cualquier valor	El segundo argumento.
Decimal (NaN)	Cualquier valor	El segundo argumento.
Decimal (no NaN)	Cualquier valor	El primer argumento.
Otro valor	Cualquier valor	El primer argumento.

newuuid()

Devuelve un UUID aleatorio de 16 bytes. Es compatible con la versión 2015-10-08 de SQL y versiones posteriores.

Ejemplo: `newuuid() = 123a4567-b89c-12d3-e456-789012345000`

numbytes(String)

Devuelve el número de bytes de la codificación UTF-8 de la cadena proporcionada. Se aplican las reglas de conversión estándar a los argumentos que no sean String. Es compatible con la versión SQL 2016-03-23 y versiones posteriores.

Ejemplos:

`numbytes("hi") = 2`

`numbytes("€") = 3`

parse_time (String, Long [, String])

Utilice la función `parse_time` para aplicar un formato legible a una marca de fecha y hora. Es compatible con la versión SQL 2016-03-23 y versiones posteriores. Para convertir una cadena de marca de hora en milisegundos, consulte [time_to_epoch \(String, String\) \(p. 613\)](#).

La `parse_time` espera los argumentos siguientes:

`pattern`

(String) Un patrón de fecha y hora que sigue el [ISO 8601](#) formato estándar. Específicamente, la función admite [Formatos Joda-Time](#).

`timestamp`

(Long) La hora que se va a formatear en milisegundos a partir del formato de hora Unix. Consulte la función [timestamp\(\)](#) (p. 613).

`timezone`

(String) La zona horaria de la fecha/hora formateada. El valor predeterminado es "UTC". La función admite [zonas horarias Joda-Time](#). Este argumento es opcional.

Ejemplos:

Cuando este mensaje se publica en el tema "A/B", la carga `{"ts": "1970.01.01 AD at 21:46:40 CST"}` se envía al bucket de 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"  
    }  
}
```

Cuando este mensaje se publica en el tema "A/B", una carga similar a `{"ts": "2017.06.09 AD at 17:19:46 UTC"}` (pero con la fecha y hora actuales) se envía al bucket de 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()` se puede usar también como una plantilla de sustitución. Por ejemplo, cuando este mensaje se publica en el tema "A/B", la carga se envía al bucket de S3 con la clave = "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:role/ROLE_NAME",
                    "bucketName": "BUCKET_NAME",
                    "key": "${parse_time('yyyy', timestamp(), 'UTC')}"
                }
            }
        ],
        "ruleName": "RULE_NAME"
    }
}
```

power(Decimal, Decimal)

Devuelve el primer argumento elevado al segundo argumento. Los argumentos `Decimal` se redondean con doble precisión antes de la aplicación de la función. Es compatible con la versión 2015-10-08 de SQL y versiones posteriores. Es compatible con la versión 2015-10-08 de SQL y versiones posteriores.

Ejemplo: `power(2, 5) = 32.0.`

Tipo de argumento 1	Tipo de argumento 2	Salida
<code>Int/Decimal</code>	<code>Int/Decimal</code>	Un <code>Decimal</code> (con doble elevado a la potencia
<code>Int/Decimal/String</code>	<code>Int/Decimal/String</code>	Un <code>Decimal</code> (con doble elevado a la potencia cadenas se convierte en un <code>Undefined</code> .
Otro valor	Otro valor	<code>Undefined</code> .

principal()

Devuelve el principal que utiliza el dispositivo para la autenticación, en función de cómo se publicó el mensaje de activación. En la siguiente tabla se describe la entidad principal devuelta para cada método y protocolo de publicación.

Cómo se publica el mensaje	Protocolo	Tipo de credenciales
Cliente MQTT	MQTT	Certificado de dispositivo
Cliente MQTT de la consola de AWS IoT	MQTT	Usuario o rol de IAM

Cómo se publica el mensaje	Protocolo	Tipo de credenciales
AWS CLI	HTTP	Usuario o rol de IAM
SDK de dispositivos de AWS IoT	MQTT	Certificado de dispositivo
SDK de dispositivos de AWS IoT	MQTT más de WebSocket	Usuario o rol de IAM

Los siguientes ejemplos muestran los distintos tipos de valores que `principal()` puede devolver:

- Huella digital del certificado X.509:
`ba67293af50bf2506f5f93469686da660c7c844e7b3950bfb16813e0d31e9373`
- ID de rol de IAM y nombre de sesión:`ABCD1EFG3HIJK2LMNOP5:my-session-name`
- devuelve un ID de usuario:`ABCD1EFG3HIJK2LMNOP5`

rand()

Devuelve un valor pseudoaleatorio, distribuido de forma uniforme entre 0,0 y 1,0. Es compatible con la versión 2015-10-08 de SQL y versiones posteriores.

Ejemplo:

`rand() = 0.8231909191640703`

regexp_matches(String, String)

Devuelve verdadero si la cadena (primer argumento) contiene una coincidencia para la expresión regular (segundo argumento).

Ejemplo:

`regexp_matches("aaaa", "a{2,}") = true.`

`regexp_matches("aaaa", "b") = false.`

Primer argumento:

Tipo de argumento	Resultado
Int	La representación String del valor Int.
Decimal	La representación String del valor Decimal.
Boolean	La representación String del valor booleano ("true" o "false").
String	El valor String.
Matriz	La representación String del valor Array (mediante reglas de conversión estándar).
Objeto	La representación String del objeto (mediante reglas de conversión estándar).
Null	Undefined.

Tipo de argumento	Resultado
Sin definir	<code>Undefined.</code>

Segundo argumento:

Tiene que ser una expresión regex válida. Los tipos que no son cadenas se convierten en valores de tipo `String` mediante reglas de conversión estándar. En función del tipo, es posible que la cadena obtenida no sea una expresión regular válida. Si el argumento (convertido) no es un regex válido, el resultado es `Undefined`.

`regexp_replace(String, String, String)`

Sustituye todos los segundos argumentos (expresiones regulares) que hay en el primer argumento por el tercer argumento. Hace referencia a los grupos de captura con `"$"`. Es compatible con la versión 2015-10-08 de SQL y versiones posteriores.

Ejemplo:

```
regexp_replace("abcd", "bc", "x") = "axd".
```

```
regexp_replace("abcd", "b(.*)d", "$1") = "ac".
```

Primer argumento:

Tipo de argumento	Resultado
<code>Int</code>	La representación <code>String</code> del valor <code>Int</code> .
<code>Decimal</code>	La representación <code>String</code> del valor <code>Decimal</code> .
<code>Boolean</code>	La representación <code>String</code> del valor booleano ("true" o "false").
<code>String</code>	El valor de origen.
Matriz	La representación <code>String</code> del valor <code>Array</code> (mediante reglas de conversión estándar).
Objeto	La representación <code>String</code> del objeto (mediante reglas de conversión estándar).
<code>Null</code>	<code>Undefined.</code>
Sin definir	<code>Undefined.</code>

Segundo argumento:

Tiene que ser una expresión regex válida. Los tipos que no son cadenas se convierten en valores de tipo `String` mediante reglas de conversión estándar. En función del tipo, es posible que la cadena obtenida no sea una expresión regular válida. Si el argumento (convertido) no es una expresión regex válida, el resultado es `Undefined`.

Tercer argumento:

Debe ser una cadena de sustitución de regex válida. (Puede hacer referencia a grupos de capturas). Los tipos que no son cadenas se convierten en valores de tipo `String` mediante reglas de conversión

estándar. Si el argumento (convertido) no es una cadena de sustitución de regex válida, el resultado es `Undefined`.

regexp_substr(String, String)

Busca la primera coincidencia del segundo parámetro (regex) en el primer parámetro. Hace referencia a los grupos de captura con `"$"`. Es compatible con la versión 2015-10-08 de SQL y versiones posteriores.

Ejemplo:

```
regexp_substr("hihihello", "hi") = "hi"
regexp_substr("hihihello", "(hi)*") = "hihi"
```

Primer argumento:

Tipo de argumento	Resultado
<code>Int</code>	La representación <code>String</code> del valor <code>Int</code> .
<code>Decimal</code>	La representación <code>String</code> del valor <code>Decimal</code> .
<code>Boolean</code>	La representación <code>String</code> del valor booleano ("true" o "false").
<code>String</code>	El argumento <code>String</code> .
<code>Matriz</code>	La representación <code>String</code> del valor <code>Array</code> (mediante reglas de conversión estándar).
<code>Objeto</code>	La representación <code>String</code> del objeto (mediante reglas de conversión estándar).
<code>Null</code>	<code>Undefined</code> .
<code>Sin definir</code>	<code>Undefined</code> .

Segundo argumento:

Tiene que ser una expresión regex válida. Los tipos que no son cadenas se convierten en valores de tipo `String` mediante reglas de conversión estándar. En función del tipo, es posible que la cadena obtenida no sea una expresión regular válida. Si el argumento (convertido) no es una expresión regex válida, el resultado es `Undefined`.

remainder(Decimal, Decimal)

Devuelve el resto de la división del primer argumento por el segundo argumento. Es igual que `mod(Decimal, Decimal)` (p. 598). También puede utilizar "%" como un operador infijo para la misma funcionalidad modulo. Es compatible con la versión 2015-10-08 de SQL y versiones posteriores.

Ejemplo: `remainder(8, 3) = 2`.

Operando izquierdo	Operando derecho	Salida
<code>Int</code>	<code>Int</code>	<code>Int</code> , el primer y el segundo argumento deben ser de tipo <code>Int</code> para ejecutar la función modulo.

Operando izquierdo	Operando derecho	Salida
Int/Decimal	Int/Decimal	Decimal, el primer argumento para los que quiere ej.
String/Int/Decimal	String/Int/Decimal	Si todas las cadenas función modulo se ejecuta en el argumento. De lo con
Otro valor	Otro valor	Undefined.

replace(Cadena, Cadena, Cadena)

Reemplaza todas las instancias del segundo argumento que hay en el primer argumento por el tercer argumento. Es compatible con la versión 2015-10-08 de SQL y versiones posteriores.

Ejemplo:

```
replace("abcd", "bc", "x") = "axd".
replace("abcdabcd", "b", "x") = "axcdaxcd".
```

Todos los argumentos

Tipo de argumento	Resultado
Int	La representación String del valor Int.
Decimal	La representación String del valor Decimal.
Boolean	La representación String del valor booleano ("true" o "false").
String	El valor de origen.
Matriz	La representación String del valor Array (mediante reglas de conversión estándar).
Objeto	La representación String del objeto (mediante reglas de conversión estándar).
Null	Undefined.
Sin definir	Undefined.

rpad(String, Int)

Devuelve el argumento de cadena, rellenado en el lado derecho con el número de espacios especificado en el segundo argumento. El argumento Int debe estar comprendido entre 0 y 1000. Si el valor proporcionado se encuentra fuera de este rango válido, el argumento se establece en el valor válido más cercano (0 o 1000). Es compatible con la versión 2015-10-08 de SQL y versiones posteriores.

Ejemplos:

```
rpad("hello", 2) = "hello ".
rpad(1, 3) = "1   ".
```

Tipo de argumento 1	Tipo de argumento 2	Resultado
<code>String</code>	<code>Int</code>	El valor <code>String</code> se rellena en el lado derecho con un número de espacios igual al valor <code>Int</code> proporcionado.
<code>String</code>	<code>Decimal</code>	El argumento <code>Decimal</code> se redondea al valor <code>Int</code> inferior más cercano y la cadena se rellena en el lado derecho con una serie de espacios igual al valor <code>Int</code> proporcionado.
<code>String</code>	<code>String</code>	El segundo argumento se convierte en un valor <code>Decimal</code> que se redondea

Tipo de argumento 1	Tipo de argumento 2	Resultado
		al valor Int inferior más cercano. El valor String se rellena en el lado derecho con un número de espacios igual al valor Int.
Otro valor	Int/Decimal/String	El primer valor se convierte en un valor de tipo String mediante las conversiones estándar y, a continuación, se aplica la función rpad a dicho valor String. Si no se puede convertir, el resultado es Undefined.
Cualquier valor	Otro valor	Undefined.

round(Decimal)

Redondea el valor `Decimal` indicado al valor `Int` más cercano. Si el valor `Decimal` está a la misma distancia de dos valores `Int`, (por ejemplo, 0,5), el valor `Decimal` se redondea al valor superior. Es compatible con la versión 2015-10-08 de SQL y versiones posteriores.

Ejemplo: `Round(1.2) = 1.`

`Round(1.5) = 2.`

`Round(1.7) = 2.`

`Round(-1.1) = -1.`

`Round(-1.5) = -2.`

Tipo de argumento	Resultado
<code>Int</code>	El argumento.
<code>Decimal</code>	El valor <code>Decimal</code> se redondea al valor <code>Int</code> inferior más cercano.
<code>String</code>	El valor <code>Decimal</code> se redondea al valor <code>Int</code> inferior más cercano. Si la cadena no se puede convertir en un valor <code>Decimal</code> , el resultado es <code>Undefined</code> .
Otro valor	<code>Undefined</code> .

rtrim(String)

Elimina todos los espacios en blanco del final (tabuladores y espacios) del valor `String` proporcionado. Es compatible con la versión 2015-10-08 de SQL y versiones posteriores.

Ejemplos:

`rtrim(" h i ") = " h i"`

Tipo de argumento	Resultado
<code>Int</code>	La representación <code>String</code> del valor <code>Int</code> .
<code>Decimal</code>	La representación <code>String</code> del valor <code>Decimal</code> .
<code>Boolean</code>	La representación <code>String</code> del valor booleano ("true" o "false").
<code>Matriz</code>	La representación <code>String</code> del valor <code>Array</code> (mediante reglas de conversión estándar).
<code>Objeto</code>	La representación <code>String</code> del objeto (mediante reglas de conversión estándar).
<code>Null</code>	<code>Undefined</code> .
<code>Sin definir</code>	<code>Undefined</code>

sign(Decimal)

Devuelve el signo del número especificado. Cuando el signo del argumento es positivo, se devuelve 1. Cuando el signo del argumento es negativo, se devuelve -1. Si el argumento es 0, se devuelve 0. Es compatible con la versión 2015-10-08 de SQL y versiones posteriores.

Ejemplos:

`sign(-7) = -1.`

`sign(0) = 0.`

`sign(13) = 1.`

Tipo de argumento	Resultado
<code>Int</code>	<code>Int</code> , el signo del valor <code>Int</code> .
<code>Decimal</code>	<code>Int</code> , el signo del valor <code>Decimal</code> .
<code>String</code>	<code>Int</code> , el signo del valor <code>Decimal</code> . La cadena se convierte en un valor <code>Decimal</code> y se devuelve el signo del valor <code>Decimal</code> . Si el valor <code>String</code> no se puede convertir en un valor <code>Decimal</code> , el resultado es <code>Undefined</code> . Es compatible con la versión 2015-10-08 de SQL y versiones posteriores.
Otro valor	<code>Undefined</code> .

sin(Decimal)

Devuelve el seno de un número en radianes. Los argumentos `Decimal` se redondean con doble precisión antes de la aplicación de la función. Es compatible con la versión 2015-10-08 de SQL y versiones posteriores.

Ejemplo: `sin(0) = 0.0`

Tipo de argumento	Resultado
<code>Int</code>	<code>Decimal</code> (con doble precisión), el seno del argumento.
<code>Decimal</code>	<code>Decimal</code> (con doble precisión), el seno del argumento.
<code>Boolean</code>	<code>Undefined</code> .
<code>String</code>	<code>Decimal</code> (con doble precisión), el seno del argumento. Si la cadena no se puede convertir en un valor <code>Decimal</code> , el resultado es <code>Undefined</code> .
<code>Matriz</code>	<code>Undefined</code> .
<code>Objeto</code>	<code>Undefined</code> .
<code>Null</code>	<code>Undefined</code> .
<code>Undefined</code>	<code>Undefined</code> .

sinh(Decimal)

Devuelve el seno hiperbólico de un número. Los valores `Decimal` se redondean con doble precisión antes de la aplicación de la función. El resultado es un valor `Decimal` de doble precisión. Es compatible con la versión 2015-10-08 de SQL y versiones posteriores.

Ejemplo: `sinh(2.3) = 4.936961805545957`

Tipo de argumento	Resultado
<code>Int</code>	<code>Decimal</code> (con doble precisión); el seno hiperbólico del argumento.
<code>Decimal</code>	<code>Decimal</code> (con doble precisión); el seno hiperbólico del argumento.
<code>Boolean</code>	<code>Undefined</code> .
<code>String</code>	<code>Decimal</code> (con doble precisión); el seno hiperbólico del argumento. Si la cadena no se puede convertir en un valor <code>Decimal</code> , el resultado es <code>Undefined</code> .
<code>Matriz</code>	<code>Undefined</code> .
<code>Objeto</code>	<code>Undefined</code> .
<code>Null</code>	<code>Undefined</code> .
<code>Sin definir</code>	<code>Undefined</code> .

substring (String, Int [, Int])

Espera un valor `String` seguido de uno o dos valores `Int`. Para un argumento `String` y un único argumento `Int`, esta función devuelve la subcadena del argumento `String` proporcionado que proviene del índice `Int` (de base 0 incluido) suministrado al final del argumento `String`. Para un argumento `String` y dos argumentos `Int`, esta función devuelve la subcadena del argumento `String` proporcionado que proviene del primer argumento de índice `Int` (de base 0 incluido) en el segundo argumento de índice `Int` (de base 0 no incluido). Los índices inferiores a cero se establecen en cero. Los índices superiores a la longitud de `String` se establecen en la longitud de `String`. Para la versión de tres argumentos, si el primer índice es superior (o igual) al segundo índice, el resultado es el valor `String` vacío.

Si los argumentos proporcionados no son (`String, Int`) ni (`String, Int, Int`), se les aplican las conversiones estándar para intentar convertirlos en los tipos adecuados. Si no es posible convertirlos, el resultado de la función será `Undefined`. Es compatible con la versión 2015-10-08 de SQL y versiones posteriores.

Ejemplos:

```
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) = "true".  
substring(false, -2.411E247) = "false".  
substring("012345", 1, 3) = "12".  
substring("012345", -50, 50) = "012345".  
substring("012345", 3, 1) = "".
```

sql_version()

Devuelve la versión de SQL especificada en esta regla. Es compatible con la versión 2015-10-08 de SQL y versiones posteriores.

Ejemplo:

```
sql_version() = "2016-03-23"
```

sqrt(Decimal)

Devuelve la raíz cuadrada de un número. Los argumentos `Decimal` se redondean con doble precisión antes de la aplicación de la función. Es compatible con la versión 2015-10-08 de SQL y versiones posteriores.

Ejemplo: `sqrt(9) = 3.0.`

Tipo de argumento	Resultado
<code>Int</code>	La raíz cuadrada del argumento.
<code>Decimal</code>	La raíz cuadrada del argumento.
<code>Boolean</code>	<code>Undefined</code> .
<code>String</code>	La raíz cuadrada del argumento. Si la cadena no se puede convertir en un valor <code>Decimal</code> , el resultado es <code>Undefined</code> .
<code>Matriz</code>	<code>Undefined</code> .
<code>Objeto</code>	<code>Undefined</code> .
<code>Null</code>	<code>Undefined</code> .
<code>Sin definir</code>	<code>Undefined</code> .

startswith(String, String)

Devuelve `Boolean`, si el primer argumento de cadena comienza con el segundo argumento de cadena. Si alguno de los argumentos es `Null` o `Undefined`, el resultado es `Undefined`. Es compatible con la versión 2015-10-08 de SQL y versiones posteriores.

Ejemplo:

```
startswith("ranger", "ran") = true
```

Tipo de argumento 1	Tipo de argumento 2	Resultado
String	String	Si la primera cadena es un número decimal, el resultado es Decimal. Si no, es Undefined.
Otro valor	Otro valor	Ambos argumentos se convierten en strings. Si la primera cadena no es un número decimal, las reglas de conversión se aplican. Si alguno de los argumentos es undefined, el resultado es Undefined.

[tan\(Decimal\)](#)

Devuelve la tangente de un número en radianes. Los valores `Decimal` se redondean con doble precisión antes de la aplicación de la función. Es compatible con la versión 2015-10-08 de SQL y versiones posteriores.

Ejemplo: `tan(3) = -0.1425465430742778`

Tipo de argumento	Resultado
<code>Int</code>	<code>Decimal</code> (con doble precisión), la tangente del argumento.
<code>Decimal</code>	<code>Decimal</code> (con doble precisión), la tangente del argumento.
<code>Boolean</code>	<code>Undefined</code> .
<code>String</code>	<code>Decimal</code> (con doble precisión), la tangente del argumento. Si la cadena no se puede convertir en un valor <code>Decimal</code> , el resultado es <code>Undefined</code> .
<code>Matriz</code>	<code>Undefined</code> .
<code>Objeto</code>	<code>Undefined</code> .
<code>Null</code>	<code>Undefined</code> .
<code>Sin definir</code>	<code>Undefined</code> .

[tanh\(Decimal\)](#)

Devuelve la tangente hiperbólica de un número en radianes. Los valores `Decimal` se redondean con doble precisión antes de la aplicación de la función. Es compatible con la versión 2015-10-08 de SQL y versiones posteriores.

Ejemplo: `tanh(2.3) = 0.9800963962661914`

Tipo de argumento	Resultado
<code>Int</code>	<code>Decimal</code> (con doble precisión), la tangente hiperbólica del argumento.
<code>Decimal</code>	<code>Decimal</code> (con doble precisión), la tangente hiperbólica del argumento.
<code>Boolean</code>	<code>Undefined</code> .

Tipo de argumento	Resultado
String	Decimal (con doble precisión), la tangente hiperbólica del argumento. Si la cadena no se puede convertir en un valor Decimal, el resultado es Undefined.
Matriz	Undefined.
Objeto	Undefined.
Null	Undefined.
Sin definir	Undefined.

time_to_epoch (String, String)

Usa `time_to_epoch` para convertir una cadena de marca de tiempo en varios milisegundos en tiempo de época Unix. Es compatible con la versión SQL 2016-03-23 y versiones posteriores. Para convertir milisegundos en una cadena de marca de hora con formato, consulte [parse_time \(String, Long \[, String\]\) \(p. 599\)](#).

La `time_to_epoch` espera los argumentos siguientes:

`timestamp`

(String) La cadena de marca de hora que se va a convertir a milisegundos a partir del formato de hora Unix. Si la cadena de marca de hora no especifica una zona horaria, la función utiliza la zona horaria UTC.

`pattern`

(String) Un patrón de fecha y hora que sigue el [ISO 8601](#) formato estándar. Específicamente, la función admite [Formatos de hora JDK11](#).

Ejemplos:

```
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()

Devuelve la marca de hora actual en milisegundos desde las 00:00:00 UTC (hora universal coordinada), jueves 1 de enero de 1970, según lo observado por el motor de reglas de AWS IoT. Es compatible con la versión 2015-10-08 de SQL y versiones posteriores.

Ejemplo: `timestamp() = 1481825251155`

topic(Decimal)

Devuelve el tema al que se ha enviado el mensaje que activó la regla. Si no se especifica ningún parámetro, se devuelve todo el tema. El parámetro `Decimal` se utiliza para especificar un segmento de tema concreto. 1 designa el primer segmento. Para el tema `foo/bar/baz`, `topic(1)` devuelve `foo`,

topic(2) devuelve bar y así sucesivamente. Es compatible con la versión 2015-10-08 de SQL y versiones posteriores.

Ejemplos:

```
topic() = "things/myThings/thingOne"
```

```
topic(1) = "things"
```

Cuando se usa [Basic Ingest \(p. 560\)](#), el prefijo inicial del tema (`$aws/rules/rule-name`) no está disponible para la función topic(). Por ejemplo, con el tema:

```
$aws/rules/BuildingManager/Buildings/Building5/Floor2/Room201/Lights
```

```
topic() = "Buildings/Building5/Floor2/Room201/Lights"
```

```
topic(3) = "Floor2"
```

traceid()

Devuelve el ID de seguimiento (UUID) del mensaje MQTT o `Undefined` si el mensaje no se ha enviado por MQTT. Es compatible con la versión 2015-10-08 de SQL y versiones posteriores.

Ejemplo:

```
traceid() = "12345678-1234-1234-1234-123456789012"
```

transform (cadena, objeto, matriz)

Devuelve una matriz de objetos que contiene el resultado de la transformación especificada `delObject`parámetro en la acción`Array`parámetro.

Es compatible con la versión SQL 2016-03-23 y versiones posteriores.

Cadena

El modo de transformación que se debe utilizar. Consulte la siguiente tabla para conocer los modos de transformación admitidos y cómo crean el `Result`desde las `Object`y `Array`parámetros.

Objeto

Objeto que contiene los atributos que se van a aplicar a cada elemento `delArray`.

Matriz

Matriz de objetos en los que se encuentran los atributos de `delObject`se aplican.

Cada objeto de esta matriz corresponde a un objeto de la respuesta de la función. Cada objeto de la respuesta de la función contiene los atributos presentes en el objeto original y los atributos proporcionados por `Object`según lo determinado por el modo de transformación especificado en `String`.

Parámetro String	Parámetro Object	Parámetro Array	Resultado
<code>enrichArray</code>	Objeto	Matriz de objetos	Matriz de objetos en la que cada objeto contiene los atributos de un elemento <code>delArray</code> parámetro y los atributos <code>delObject</code> parámetro.

Parámetro String	Parámetro Object	Parámetro Array	Resultado
Cualquier otro valor	Cualquier valor	Cualquier valor	Sin definir

Note

La matriz devuelta por esta función está limitada a 128 KiB.

Ejemplo 1 de función de transformación

Este ejemplo le muestra cómo se muestra la acción `transform()` produce una única matriz de objetos a partir de un objeto de datos y de una matriz.

En este ejemplo, se publica el siguiente mensaje en el tema MQTT A/B.

```
{
  "attributes": {
    "data1": 1,
    "data2": 2
  },
  "values": [
    {
      "a": 3
    },
    {
      "b": 4
    },
    {
      "c": 5
    }
  ]
}
```

Esta instrucción SQL para una acción de regla de tema utiliza la función `transform()` con `unStringValue` y `enrichArray`. En este ejemplo, los objetos `values` y `attributes` de la carga útil del mensaje y `array` contiene tres objetos.

```
select value transform("enrichArray", attributes, values) from 'A/B'
```

Al recibir la carga útil del mensaje, la instrucción SQL se evalúa según la siguiente respuesta.

```
[
  {
    "a": 3,
    "data1": 1,
    "data2": 2
  },
  {
    "b": 4,
    "data1": 1,
    "data2": 2
  },
  {
    "c": 5,
    "data1": 1,
    "data2": 2
  }
]
```

Ejemplo de función de transformación 2

Este ejemplo le muestra cómo se muestra la acción `transform()` puede utilizar valores literales para incluir y cambiar el nombre de atributos individuales de la carga útil del mensaje.

En este ejemplo, se publica el siguiente mensaje en el tema `MQTTA/B`. Es el mismo mensaje que se utilizó en [the section called “Ejemplo 1 de función de transformación” \(p. 615\)](#).

```
{  
    "attributes": {  
        "data1": 1,  
        "data2": 2  
    },  
    "values": [  
        {  
            "a": 3  
        },  
        {  
            "b": 4  
        },  
        {  
            "c": 5  
        }  
    ]  
}
```

Esta instrucción SQL para una acción de regla de tema utiliza la función `transform()` con un `string` valor de `enrichArray`. La función `transform()` tiene un único atributo denominado `key` con el valor `data1` en la carga útil del mensaje y `array` es el `values` matriz, que contiene los mismos tres objetos utilizados en el ejemplo anterior.

```
select value transform("enrichArray", {"key": attributes.data1}, values) from 'A/B'
```

Al recibir la carga útil del mensaje, esta instrucción SQL se evalúa según la siguiente respuesta. Observe cómo la propiedad `data1` se llama `key` en la respuesta.

```
[  
    {  
        "a": 3,  
        "key": 1  
    },  
    {  
        "b": 4,  
        "key": 1  
    },  
    {  
        "c": 5,  
        "key": 1  
    }  
]
```

Ejemplo de función de transformación 3

Este ejemplo le muestra cómo se muestra la acción `transform()` se puede utilizar en cláusulas `SELECT` anidadas para seleccionar varios atributos y crear nuevos objetos para su posterior procesamiento.

En este ejemplo, se publica el siguiente mensaje en el tema `MQTTA/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
        }
    ]
}
```

La `Object` para esta función de transformación es el objeto devuelto por la instrucción SELECT, que contiene los elementos del mensaje `data2` objeto. La `Array`: consta de los dos objetos de `data2.cMatriz` en el mensaje original.

```
select value transform('enrichArray', (select a, b from data2), (select value c from data2)) from 'A/B'
```

Con el mensaje anterior, la instrucción SQL se evalúa según la siguiente respuesta.

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

La matriz devuelta en esta respuesta podría utilizarse con acciones de reglas de tema que admiten `batchMode`.

trim(String)

Elimina todos los espacios en blanco del principio y del final del valor `String` proporcionado. Es compatible con la versión 2015-10-08 de SQL y versiones posteriores.

Ejemplo:

`Trim(" hi ") = "hi"`

Tipo de argumento	Resultado
<code>Int</code>	La representación <code>String</code> de <code>Int</code> con todos los espacios en blanco del principio y del final suprimidos.
<code>Decimal</code>	La representación <code>String</code> de <code>Decimal</code> con todos los espacios en blanco del principio y del final suprimidos.
<code>Boolean</code>	La representación <code>String</code> de <code>Boolean</code> ("true" o "false") con todos los espacios en blanco del principio y del final suprimidos.
<code>String</code>	El argumento <code>String</code> con todos los espacios en blanco del principio y del final suprimidos.
<code>Matriz</code>	La representación <code>String</code> del valor <code>Array</code> mediante reglas de conversión estándar.
<code>Objeto</code>	La representación <code>String</code> del objeto mediante reglas de conversión estándar.
<code>Null</code>	<code>Undefined</code> .
<code>Sin definir</code>	<code>Undefined</code> .

trunc(Decimal, Int)

Trunca el primer argumento según el número del valor `Decimal` especificado por el segundo argumento. Si el segundo argumento es inferior a cero, se establece en cero. Si el segundo argumento es superior a 34, se establece en 34. Los ceros del final se eliminan del resultado. Es compatible con la versión 2015-10-08 de SQL y versiones posteriores.

Ejemplos:

```
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	Resultado
<code>Int</code>	<code>Int</code>	El valor de origen.
<code>Int/Decimal</code>	<code>Int/Decimal</code>	El primer argumento si el segundo argumento es un valor <code>Int</code> , se redondea al número de dígitos especificados.
<code>Int/Decimal/String</code>	<code>Int/Decimal</code>	El primer argumento si el segundo argumento es un valor <code>Int</code> , se redondea al número de dígitos especificados. Los valores de tipo <code>String</code> se convierten a tipo <code>Decimal</code> . Si se produce una excepción, el resultado es <code>Undefined</code> .

Tipo de argumento 1	Tipo de argumento 2	Resultado
Otro valor		Undefined.

upper(String)

Muestra la versión en mayúsculas del valor `String` indicado. Los argumentos que no sean `String` se convierten en `String` mediante las reglas de conversión estándar. Es compatible con la versión 2015-10-08 de SQL y versiones posteriores.

Ejemplos:

```
upper("hello") = "HELLO"
upper(["hello"]) = ["\"HELLO\"]"
```

Literales

Puede especificar directamente objetos literales en las cláusulas SELECT y WHERE de su regla SQL, lo que puede ser útil para pasar información.

Note

Los literales solo están disponibles cuando se utiliza SQL versión 2016-03-23 o versiones posteriores.

Se utiliza una sintaxis de objeto JSON (pares clave-valor separados con comas, donde las claves son cadenas y los valores son de tipo JSON escritos entre llaves {}). Por ejemplo:

Carga de entrada publicada en el tema `topic/subtopic: {"lat_long": [47.606,-122.332]}`

Instrucción SQL: `SELECT {'latitude': get(lat_long, 0), 'longitude':get(lat_long, 1)} as lat_long FROM 'topic/subtopic'`

La carga de salida obtenida sería: `{"lat_long": {"latitude": 47.606, "longitude": -122.332}}`.

También puede especificar directamente matrices en las cláusulas SELECT y WHERE de su regla SQL, lo que le permite agrupar información. Se utiliza una sintaxis JSON (elementos separados con comas entre corchetes [] para crear un literal de Array). Por ejemplo:

Carga de entrada publicada en el tema `topic/subtopic: {"lat": 47.696, "long": -122.332}`

Instrucción SQL: `SELECT [lat,long] as lat_long FROM 'topic/subtopic'`

La carga de salida obtenida sería: `{"lat_long": [47.606,-122.332]}`.

Instrucciones case

Las instrucciones de caso se pueden utilizar para ejecutar bifurcaciones, como una instrucción switch.

Sintaxis:

```
CASE v WHEN t[1] THEN r[1]
WHEN t[2] THEN r[2] ...
WHEN t[n] THEN r[n]
ELSE r[e] END
```

La expresión `vse` evalúa y se compara con todas las `t[i]` valor de cada uno `WHEN` cláusula. Si se encuentra una coincidencia, la correspondiente `x[i]` expresión se convierte en el resultado de la `CASE` statement. Las `WHEN` cláusulas se evalúan en orden de modo que, si hay más de una cláusula coincidente, el resultado de la primera cláusula de coincidencia se convierte en el resultado de la `CASE` statement. Si no hay coincidencias, `r[e]` de la `ELSE` cláusula es el resultado. Si no hay coincidencia y no `ELSE` cláusula, el resultado es `undefined`.

`CASE` las instrucciones requieren como mínimo una `WHEN` cláusula. Una `ELSE` cláusula es opcional.

Por ejemplo:

Carga de entrada publicada en el tema `topic/subtopic`:

```
{  
    "color": "yellow"  
}
```

Instrucción 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'
```

La carga de salida obtenida sería:

```
{  
    "instructions": "caution"  
}
```

Note

Si `vse` es `undefined`, el resultado de la exposición de un caso es `undefined`.

Extensiones JSON

Puede utilizar las extensiones siguientes de la sintaxis ANSI SQL para facilitar el trabajo con objetos JSON anidados.

`."` Operador

Este operador tiene acceso a los miembros de los objetos y las funciones JSON integrados, igual que en ANSI SQL y JavaScript. Por ejemplo:

```
SELECT foo.bar AS bar.baz FROM 'topic/subtopic'
```

selecciona el valor de la `bar` propiedad en el `foo` objeto de la siguiente carga útil de mensajes enviada a la `topic/subtopic` tema.

```
{  
    "foo": {  
        "bar": "RED",  
        "bar1": "GREEN",  
        "bar2": "BLUE"  
    }  
}
```

```
}
```

Si el nombre de una propiedad JSON incluye un guión o caracteres numéricos, la simple notación «punto» no funcionará. En su lugar, debe utilizar la acción[obtener la función \(p. 589\)](#)para extraer el valor de la propiedad.

En este ejemplo se envía el mensaje siguiente aliot/rules tema.

```
{
  "mydata": {
    "item2": {
      "0": {
        "my-key": "myValue"
      }
    }
  }
}
```

Normalmente, el valor de my-key se identificaría como en esta consulta.

```
SELECT * from iot/rules WHERE mydata.item2.0.my-key= "myValue"
```

Sin embargo, porque el nombre de la propiedad my-key contiene un guión y item2 contiene un carácter numérico, el[obtener la función \(p. 589\)](#)debe utilizarse como se muestra en la siguiente consulta.

```
SELECT * from 'iot/rules' WHERE get(get(get(mydata,"item2"),"0"),"my-key") = "myValue"
```

Operador *

Funciona igual que el comodín * en ANSI SQL. Solo se utiliza en la cláusula SELECT y crea un objeto JSON nuevo que contiene los datos del mensaje. Si la carga del mensaje no tiene el formato JSON, * devuelve toda la carga del mensaje como bytes sin procesar. Por ejemplo:

```
SELECT * FROM 'topic/subtopic'
```

Aplicación de una función a un valor de atributo

A continuación, se muestra un ejemplo de carga JSON que podría publicar un dispositivo:

```
{
  "deviceid" : "iot123",
  "temp" : 54.98,
  "humidity" : 32.43,
  "coords" : {
    "latitude" : 47.615694,
    "longitude" : -122.3359976
  }
}
```

En el ejemplo siguiente se aplica una función a un valor de atributo de una carga JSON:

```
SELECT temp, md5(deviceid) AS hashed_id FROM topic/#
```

El resultado de esta consulta es el objeto JSON siguiente:

```
{
```

```
    "temp": 54.98,  
    "hashed_id": "e37f81fb397e595c4aeb5645b8cbbbd1"  
}
```

Plantillas de sustitución

Puede utilizar una plantilla de sustitución para aumentar los datos JSON que se devuelven cuando se activa una regla y AWS IoT realiza una acción. La sintaxis de una plantilla de sustitución es `${expression}` , donde `expression` puede ser cualquier expresión compatible con AWS IoT en las cláusulas `SELECT` o `WHERE` y en [Acciones de reglas de AWS IoT \(p. 479\)](#). Esta expresión se puede conectar a un campo de acción de una regla, lo que le permite configurar dinámicamente una acción. En efecto, esta función sustituye a una parte de información de una acción. Esto incluye funciones, operadores e información presente en la carga del mensaje original.

Important

Dado que las expresiones en plantillas de sustitución se evalúan por separado de la declaración `"SELECT..."` , no se puede hacer referencia a un alias creado con la cláusula `AS` . Solo puede hacer referencia a la información presente en la carga original. [funciones \(p. 575\)](#), [y Operadores \(p. 569\)](#).

Para obtener más información acerca de las expresiones admitidas, consulte [Referencia de la SQL de AWS IoT \(p. 562\)](#).

Las siguientes acciones de las reglas admiten plantillas de sustitución. Cada acción admite diferentes campos que se pueden sustituir.

- [Apache Kafka \(p. 481\)](#)
- [Alarms de CloudWatch \(p. 489\)](#)
- [CloudWatch Logs \(p. 490\)](#)
- [Métricas de CloudWatch \(p. 491\)](#)
- [DynamoDB \(p. 493\)](#)
- [DynamoDBv2 \(p. 495\)](#)
- [Elasticsearch \(p. 497\)](#)
- [HTTP \(p. 498\)](#)
- [IoT Analytics \(p. 525\)](#)
- [IoT Events \(p. 527\)](#)
- [IoT SiteWise \(p. 528\)](#)
- [Kinesis Data Streams \(p. 534\)](#)
- [Kinesis Data Firehose \(p. 532\)](#)
- [Lambda \(p. 536\)](#)
- [OpenSearch \(p. 538\)](#)
- [Republish \(p. 540\)](#)
- [S3 \(p. 541\)](#)
- [SNS \(p. 543\)](#)
- [SQS \(p. 545\)](#)
- [Step Functions \(p. 547\)](#)
- [Timestream \(p. 548\)](#)

Las plantillas de sustitución aparecen en los parámetros de acción dentro de una regla:

```
{
```

```
"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"  
        }  
    }  
]
```

Si esta regla se activa mediante el siguiente JSON publicado en `my/iot/topic`:

```
{  
    "deviceid": "iot123",  
    "temp": 54.98,  
    "humidity": 32.43,  
    "coords": {  
        "latitude": 47.615694,  
        "longitude": -122.3359976  
    }  
}
```

Esta regla publica el siguiente JSON en `my/iot/topic/republish`, que AWS IoT utiliza para sustituir `${topic()}/republish`:

```
{  
    "deviceid": "iot123",  
    "temp": 54.98,  
    "humidity": 32.43,  
    "coords": {  
        "latitude": 47.615694,  
        "longitude": -122.3359976  
    },  
    "timestamp": 1579637878451  
}
```

Consultas de objetos anidados

Puede utilizar cláusulas SELECT anidadas para consultar atributos dentro de matrices y objetos JSON internos. Es compatible con la versión SQL 2016-03-23 y versiones posteriores.

Considere el siguiente mensaje 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

Puede convertir valores en una nueva matriz con la siguiente regla.

```
SELECT (SELECT VALUE n FROM e) as sensors FROM 'my/topic'
```

La regla generará la salida siguiente.

```
{  
    "sensors": [  
        "temperature",  
        "light",  
        "acidity"  
    ]  
}
```

Example

Usando el mismo mensaje MQTT, también puede consultar un valor específico dentro de un objeto anidado con la siguiente regla.

```
SELECT (SELECT v FROM e WHERE n = 'temperature') as temperature FROM 'my/topic'
```

La regla generará la salida siguiente.

```
{  
    "temperature": [  
        {  
            "v": 22.5  
        }  
    ]  
}
```

Example

También puede aplinar la salida con una regla más complicada.

```
SELECT get((SELECT v FROM e WHERE n = 'temperature'), 0).v as temperature FROM 'topic'
```

La regla generará la salida siguiente.

```
{  
    "temperature": 22.5  
}
```

Uso de las cargas binarias

Cuando la carga del mensaje deba tratarse como datos binarios sin procesar, (en lugar de como un objeto JSON), puede utilizar el operador * para hacer referencia a ella en una cláusula SELECT. Esto funciona para cargas de trabajo que no sean de JSON con algunas acciones de regla, como la [acción S3](#).

Ejemplos de carga binaria

Si utiliza * para hacer referencia a la carga del mensaje como datos binarios sin procesar, puede agregar datos a la regla. Si tiene una carga útil vacía o JSON, se pueden agregar datos a la carga útil resultante mediante la regla. A continuación se muestran ejemplos de compatibilidad SELECT Cláusulas.

- Puede utilizar las siguientes SELECT cláusulas con solo un * para cargas útiles binarias.

- ```
SELECT * FROM 'topic/subtopic'
```

- ```
SELECT * FROM 'topic/subtopic' WHERE timestamp() % 12 = 0
```

- También puede agregar datos y utilizar lo siguientes SELECT Cláusulas.

```
• SELECT *, principal() as principal, timestamp() as time FROM 'topic/subtopic'  
• SELECT encode(*, 'base64') AS data, timestamp() AS ts FROM 'topic/subtopic'
```

- También puede utilizar estas SELECT cláusulas con cargas binarias.

- Lo siguiente hace referencia a device_type en la cláusula WHERE.

```
SELECT * FROM 'topic/subtopic' WHERE device_type = 'thermostat'
```

- También se admite lo siguiente.

```
{  
    "sql": "SELECT * FROM 'topic/subtopic'"  
    "actions": [  
        {"  
            "republish": {  
                "topic": "device/${device_id}"  
            }  
        }  
    ]  
}
```

Las siguientes acciones de regla no admiten cargas útiles binarias, por lo que debe decodificarlas.

- Algunas acciones de regla no admiten la entrada de carga útil binaria, como un [Acción Lambda](#), por lo que debe decodificar las cargas útiles binarias. La acción de regla de Lambda puede recibir datos binarios, si está codificado en base64 y en una carga JSON. Puede hacer esto cambiando la regla a la siguiente.

```
SELECT encode(*, 'base64') AS data FROM 'my_topic'
```

- La instrucción SQL no admite cadena como entrada. Para convertir una entrada de cadena de en JSON, puede ejecutar el comando siguiente.

```
SELECT decode(encode(*, 'base64'), 'base64') AS payload FROM 'topic'
```

Versiones de SQL

El motor de reglas de AWS IoT utiliza una sintaxis similar a SQL para seleccionar los datos de los mensajes de MQTT. Las instrucciones SQL se interpretan según la versión de SQL especificada en la propiedad awsIotSqlVersion de un documento JSON que describe la regla. Para obtener más información acerca de la estructura de los documentos de reglas JSON, consulte [Creación de una regla \(p. 475\)](#). La propiedad awsIotSqlVersion le permite especificar la versión del motor de reglas SQL de AWS IoT que desea utilizar. Cuando se implementa una nueva versión, puede continuar utilizando una versión anterior o cambiar la regla para utilizar la nueva versión. Las reglas actuales seguirán utilizando la versión con la que se crearon.

En el siguiente ejemplo de JSON se muestra cómo especificar la versión de SQL mediante la propiedad awsIotSqlVersion:

```
{  
    "sql": "expression",  
    "ruleDisabled": false,  
    "awsIotSqlVersion": "2016-03-23",  
    "actions": [{  
        "topic": "device/test"  
    }]  
}
```

```
        "republish": {
            "topic": "my-mqtt-topic",
            "roleArn": "arn:aws:iam::123456789012:role/my-iot-role"
        }
    }]
}
```

AWS IoT admite actualmente las siguientes versiones de SQL:

- 2016-03-23— La versión de SQL creada el 23/03/2016 (recomendado).
- 2015-10-08— La versión de SQL original creada el 08/10/2015.
- beta: la versión beta de SQL más reciente. Esta versión podría introducir cambios bruscos en sus reglas.

Novedades de la versión del motor de reglas SQL del 23/03/2016

- Soluciones para seleccionar objetos JSON anidados.
- Soluciones para consultas de matriz.
- Compatibilidad con consultas dentro de objetos. Para obtener más información, consulte [Consultas de objetos anidados \(p. 623\)](#).
- Compatibilidad con la generación de una matriz como objeto de nivel superior.
- Adición de la función `encode(value, encodingScheme)`, que se puede aplicar en datos con formato JSON y no JSON. Para obtener más información, consulte la [función de codificación \(p. 588\)](#).

Generación de Array como objeto de nivel superior

Esta característica permite que una regla devuelva una matriz como objeto de nivel superior. Por ejemplo, si se recibe el mensaje MQTT siguiente:

```
{
    "a": {"b": "c"},
    "arr": [1, 2, 3, 4]
}
```

Y la regla siguiente:

```
SELECT VALUE arr FROM 'topic'
```

La regla generará la salida siguiente.

```
[1, 2, 3, 4]
```

Servicio Device Shadow de AWS IoT

El servicio Device Shadow de AWS IoT agrega sombras a objetos de AWS IoT. Las sombras pueden hacer que el estado de un dispositivo esté disponible para las aplicaciones y otros servicios, independientemente de que el dispositivo esté conectado a AWS IoT o no. Los objetos de AWS IoT pueden tener varias sombras con nombre para que su solución de IoT tenga más opciones para conectar los dispositivos a otras aplicaciones y servicios.

AWS IoT los objetos no tienen sombras con nombre hasta que se crean explícitamente; sin embargo, se crea una sombra clásica sin nombre para un objeto cuando se crea la objeto. Las sombras se pueden crear, actualizar y eliminar mediante la consola de AWS IoT. Los dispositivos, otros clientes web y los servicios pueden crear, actualizar y eliminar sombras mediante MQTT y la [Temas MQTT reservados \(p. 113\)](#), HTTP mediante el [API REST de sombra de dispositivo \(p. 653\)](#), y el [AWS CLI para AWS IoT](#). Porque las sombras se almacenan en la nube, pueden recopilar y notificar datos de estado del dispositivo desde aplicaciones y otros servicios en la nube, independientemente de si el dispositivo está conectado o no.

Uso de sombras

Las sombras proporcionan un almacén de datos de confianza para dispositivos, aplicaciones y otros servicios en la nube para compartir datos. Permiten que dispositivos, aplicaciones y otros servicios en la nube se conecten y desconecten sin perder el estado de un dispositivo.

Mientras los dispositivos, las aplicaciones y otros servicios en la nube están conectados a AWS IoT, pueden acceder y controlar el estado actual de un dispositivo a través de sus sombras. Por ejemplo, una aplicación puede solicitar un cambio en el estado de un dispositivo actualizando una sombra. AWS IoT publica un mensaje que indica el cambio en el dispositivo. El dispositivo recibe este mensaje, actualiza su estado para que coincida y publica un mensaje con su estado actualizado. El servicio Device Shadow refleja este estado actualizado en la sombra correspondiente. La aplicación puede suscribirse a la actualización de la sombra o puede consultar la sombra para conocer su estado actual.

Cuando un dispositivo se desconecta, una aplicación puede seguir comunicándose con AWS IoT y las sombras del dispositivo. Cuando el dispositivo se vuelve a conectar, recibe el estado actual de sus sombras para que pueda actualizar su estado para que coincida con el de sus sombras y, a continuación, publicar un mensaje con su estado actualizado. Del mismo modo, cuando una aplicación se desconecta y el estado del dispositivo cambia mientras está fuera de línea, el dispositivo mantiene la sombra actualizada para que la aplicación pueda consultar las sombras para conocer su estado actual cuando se vuelva a conectar.

Si los dispositivos están desconectados con frecuencia y desea configurar los dispositivos para que reciban mensajes delta después de volver a conectarse, puede utilizar la función de sesión persistente. Para obtener más información acerca del período de caducidad de la sesión persistente, consulte [Período de caducidad de la sesión persistente](#).

Elegir utilizar sombras con nombre o sin nombre

El servicio Device Shadow admite sombras clásicas con nombre y sin nombre, como se utilizaba en el pasado. Un objeto puede tener varias sombras con nombre y no más de una sombra clásica sin nombre. Un objeto puede tener sombras con nombre y sin nombre al mismo tiempo; sin embargo, la API utilizada para acceder a cada una es ligeramente diferente, por lo que podría ser más eficiente decidir qué tipo de sombra funcionaría mejor para su solución y usar solo dicho tipo. Para obtener más información acerca de la API para acceder a las sombras, consulte [Temas de sombra \(p. 113\)](#).

Mediante las sombras con nombre, puede crear distintas vistas del estado de un objeto. Por ejemplo, podría dividir un objeto con muchas propiedades en sombras con grupos lógicos de propiedades, cada una identificada por su nombre de sombra. También puede limitar el acceso a las propiedades agrupándolas en distintas sombras y utilizando políticas para controlar el acceso. Para obtener más información acerca de las políticas que se deben usar con sombras de dispositivos, consulte [Acciones, recursos y claves de condición para AWS IoT](#).

Las sombras clásicas sin nombre son más sencillas, pero algo más limitadas que las sombras con nombre. Cada objeto de AWS IoT puede tener solo una sombra sin nombre. Si espera que la solución de IoT tenga una necesidad limitada de datos de sombra, puede que así sea como desee comenzar a usar sombras. Sin embargo, si cree que es posible que desee agregar sombras adicionales en el futuro, plantéese la posibilidad de utilizar sombras con nombre desde el principio.

La indexación de flotas admite sombras sin nombre y sombras con nombre de forma diferente. Para obtener más información, consulte [Indexación de flota \(p. 825\)](#).

Acceso a sombras

Cada sombra tiene un tema MQTT (p. 113) reservado y una URL HTTP (p. 653) que admite las acciones `get`, `update` y `delete` en la sombra.

Las sombras utilizan [documentos de sombra JSON \(p. 665\)](#) para almacenar y recuperar datos. Un documento de sombra contiene una propiedad de estado que describe estos aspectos del estado del dispositivo:

- `desired`

Las aplicaciones especifican los estados deseados de las propiedades del dispositivo actualizando el objeto `desired`.

- `reported`

Los dispositivos notifican su estado actual en el objeto `reported`.

- `delta`

AWS IoT notifica las diferencias entre el estado deseado y el notificado en el objeto `delta`.

Los datos almacenados en una sombra están determinados por la propiedad de estado del cuerpo del mensaje de la acción de actualización. Las acciones de actualización posteriores pueden modificar los valores de un objeto de datos existente y también agregar y eliminar claves y otros elementos del objeto de estado de la sombra. Para obtener más información sobre cómo acceder a las sombras, consulte [Uso de sombras en dispositivos \(p. 631\)](#) y [Uso de sombras en aplicaciones y servicios \(p. 634\)](#).

Important

El permiso para realizar solicitudes de actualización debe limitarse a aplicaciones y dispositivos de confianza. Esto evita que la propiedad de estado de la sombra se cambie de forma inesperada; de lo contrario, los dispositivos y aplicaciones que usan la sombra deben diseñarse para esperar que cambien las claves de la propiedad de estado.

Uso de sombras en dispositivos, aplicaciones y otros servicios en la nube

El uso de sombras en dispositivos, aplicaciones y otros servicios en la nube requiere coherencia y coordinación entre todos ellos. El servicio Device Shadow de AWS IoT almacena el estado de la sombra, envía mensajes cuando cambia el estado de la sombra y responde a los mensajes que cambian su estado.

Los dispositivos, las aplicaciones y otros servicios en la nube de la solución IoT deben administrar su estado y mantenerlo coherente con el estado de la sombra del dispositivo.

Los datos de estado de sombra son dinámicos y los pueden modificar los dispositivos, las aplicaciones y otros servicios en la nube con permiso para acceder a la sombra. Por esta razón, es importante considerar cómo interactuarán con la sombra cada dispositivo, aplicación y otro servicio en la nube. Por ejemplo:

- Los dispositivos deben escribir solo en la propiedad `reported` del estado de la sombra al comunicar datos de estado a la sombra.
- Las aplicaciones y otros servicios en la nube deben escribir solo en la propiedad `desired` al comunicar solicitudes de cambio de estado al dispositivo a través de la sombra.

Important

Los datos contenidos en un objeto de datos de sombra son independientes de los de otras sombras y otras propiedades de objetos, como los atributos de un objeto y el contenido de los mensajes MQTT que el dispositivo de un objeto podría publicar. Sin embargo, un dispositivo puede notificar los mismos datos en diferentes temas y sombras de MQTT si es necesario.

Un dispositivo que admite varias sombras debe mantener la coherencia de los datos que notifica en las distintas sombras.

Orden de los mensajes

No se garantiza que los mensajes generados por el servicio de AWS IoT lleguen al dispositivo siguiendo un orden específico. La siguiente situación muestra lo que sucede en este caso.

Documento de estado inicial:

```
{  
  "state": {  
    "reported": {  
      "color": "blue"  
    }  
  },  
  "version": 9,  
  "timestamp": 123456776  
}
```

Actualización 1:

```
{  
  "state": {  
    "desired": {  
      "color": "RED"  
    }  
  },  
  "version": 10,  
  "timestamp": 123456777  
}
```

Actualización 2:

```
{  
  "state": {  
    "desired": {  
      "color": "GREEN"  
    }  
  }  
}
```

```
    },
    "version": 11,
    "timestamp": 123456778
}
```

Documento de estado final:

```
{
  "state": {
    "reported": {
      "color": "GREEN"
    }
  },
  "version": 12,
  "timestamp": 123456779
}
```

Se obtienen dos mensajes delta:

```
{
  "state": {
    "color": "RED"
  },
  "version": 11,
  "timestamp": 123456778
}
```

```
{
  "state": {
    "color": "GREEN"
  },
  "version": 12,
  "timestamp": 123456779
}
```

El dispositivo puede recibir estos mensajes de forma desordenada. Dado que el estado de estos mensajes es acumulable, un dispositivo puede descartar con toda seguridad todos los mensajes cuyo número de versión sea anterior a la del mensaje del cual se hace un seguimiento. Si el dispositivo recibe el delta de la versión 12 antes que el de la versión 11, puede descartar sin problemas el mensaje de la versión 11.

Recorte de mensajes de sombra

Para reducir el tamaño de los mensajes de sombra que se envían al dispositivo, defina una regla que seleccione solo los campos que necesita el dispositivo y que después vuelva a publicar el mensaje en un tema MQTT al que el dispositivo esté escuchando.

La regla se especifica en JSON y debe tener el aspecto siguiente:

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

}

La instrucción SELECT determina qué campos del mensaje se volverán a publicar en el tema especificado. Se usa el comodín "+" para seleccionar todos los nombres de sombra. La regla especifica que todos los mensajes coincidentes deben volver a publicarse en el tema especificado. En tal caso, la función "topic()" se utiliza para especificar el tema en el que se vuelve a publicar. topic(3) toma el valor del nombre de objeto del tema original. Para obtener más información sobre la creación de reglas, consulte [Reglas para AWS IoT \(p. 472\)](#).

Uso de sombras en dispositivos

En esta sección se describen las comunicaciones de dispositivos con sombras mediante mensajes MQTT, el método preferido para que los dispositivos se comuniquen con el servicio Device Shadow de AWS IoT.

Las comunicaciones de sombra emulan un modelo de solicitud/respuesta utilizando el modelo de comunicación de publicación/suscripción de MQTT. Cada acción de sombra consta de un tema de solicitud, un tema de respuesta correcta (accepted) y un tema de respuesta de error (rejected).

Si desea que las aplicaciones y servicios puedan determinar si un dispositivo está conectado, consulte [Detección de un dispositivo conectado \(p. 635\)](#).

Important

Dado que MQTT utiliza un modelo de comunicación de publicación/suscripción, debe suscribirse a los temas de respuesta para un tema de solicitud. Si no lo hace, es posible que no reciba la respuesta a la solicitud que publique.

Si usa un [SDK para dispositivos con AWS IoT \(p. 1274\)](#) para llamar a las API del servicio Device Shadow, esto se gestiona por ti.

En los ejemplos de esta sección se utiliza una forma abreviada del tema donde se utiliza la **ShadowTopicPrefix**. Puede hacer referencia a una sombra con nombre o sin nombre, tal como se describe en esta tabla.

Las sombras pueden ser con nombre o sin nombre (clásico). Los temas utilizados por cada uno solo difieren en el prefijo del tema. Esta tabla muestra el prefijo de tema utilizado por cada tipo de sombra.

Valor ShadowTopicPrefix	Tipo de sombra
\$aws/things/ thingName /shadow	Sombra sin nombre (clásica)
\$aws/things/ thingName /shadow/ name / shadowName	Sombra con nombre

Important

Asegúrese de que el uso de las sombras por parte de la aplicación o servicio sea coherente y compatible con las implementaciones correspondientes en los dispositivos. Por ejemplo, tenga en cuenta cómo se crean, actualizan y eliminan las sombras. Tenga en cuenta también cómo se tratan las actualizaciones en el dispositivo y en las aplicaciones o servicios que acceden al dispositivo a través de una sombra. El diseño debe ser claro respecto a cómo se actualiza y notifica el estado del dispositivo y cómo interactúan las aplicaciones y los servicios con el dispositivo y sus sombras.

Para crear un tema completo, seleccione el **ShadowTopicPrefix** para el tipo de sombra al que desea hacer referencia, reemplace **thingName** y **shadowName** si procede, con sus valores correspondientes y,

a continuación, anexe el código auxiliar del tema como se muestra en la tabla siguiente. Recuerde que los temas distinguen entre mayúsculas y minúsculas.

Consulte [Temas de sombra \(p. 113\)](#) para obtener más información acerca de los temas reservados para las sombras.

Inicialización del dispositivo en la primera conexión a AWS IoT

Después de que un dispositivo se registre con AWS IoT, debe suscribirse a estos mensajes MQTT para las sombras que admite.

Tema	Significado	Acción que debe realizar un dispositivo cuando se recibe este tema
<i>ShadowTopicPrefix/delete/accepted</i>	Se aceptó la delete solicitud e AWS IoT eliminó la sombra.	Las acciones necesarias para incorporar la sombra eliminada, como detener la publicación de actualizaciones.
<i>ShadowTopicPrefix/delete/rejected</i>	AWS IoT rechazó la solicitud delete y la sombra no se eliminó. El cuerpo del mensaje contiene la información de error.	Responda al mensaje de error en el cuerpo del mensaje.
<i>ShadowTopicPrefix/get/accepted</i>	AWS IoT aceptó la solicitud get y el cuerpo del mensaje contiene el documento de sombra actual.	Las acciones necesarias para procesar el documento de estado en el cuerpo del mensaje.
<i>ShadowTopicPrefix/get/rejected</i>	AWS IoT rechazó la solicitud get y el cuerpo del mensaje contiene la información de error.	Responda al mensaje de error en el cuerpo del mensaje.
<i>ShadowTopicPrefix/update/accepted</i>	AWS IoT aceptó la solicitud update y el cuerpo del mensaje contiene el documento de sombra actual.	Confirme que los datos actualizados en el cuerpo del mensaje coinciden con el estado del dispositivo.
<i>ShadowTopicPrefix/update/rejected</i>	AWS IoT rechazó la solicitud update y el cuerpo del mensaje contiene la información de error.	Responda al mensaje de error en el cuerpo del mensaje.
<i>ShadowTopicPrefix/update/delta</i>	El documento de sombra se actualizó mediante una solicitud a AWS IoT y el cuerpo del mensaje contiene los cambios solicitados.	Actualice el estado del dispositivo para que coincida con el estado deseado en el cuerpo del mensaje.
<i>ShadowTopicPrefix/update/documents</i>	Recientemente se completó una actualización de la sombra y el cuerpo del mensaje contiene el documento de sombra actual.	Confirme que el estado actualizado en el cuerpo del mensaje coincide con el estado del dispositivo.

Después de suscribirse a los mensajes de la tabla anterior para cada sombra, el dispositivo debe probar si las sombras que admite ya se han creado publicando un tema /get en cada sombra. Si se recibe un

mensaje `/get/accepted`, el cuerpo del mensaje contiene el documento de sombra, que el dispositivo puede utilizar para inicializar su estado. Si se recibe un mensaje `/get/rejected`, la sombra debe crearse publicando un mensaje `/update` con el estado actual del dispositivo.

Supongamos que tiene un objeto `My_IoT_Thing` que no tiene sombras clásicas o con nombre. Si publicas ahora un `/get` sobre el tema reservado `$aws/things/My_IoT_Thing/shadow/get`, devuelve un error en la `$aws/things/My_IoT_Thing/shadow/get/rejected` porque la cosa no tiene sombras. Para resolver este error, publique primero un `/update` mediante el tema `$aws/things/My_IoT_Thing/shadow/update` con el estado actual del dispositivo, como la siguiente carga útil.

```
"state": {  
    "reported": {  
        "welcome": "aws-iot",  
        "color": "yellow"  
    }  
}
```

Ahora se crea una sombra clásica para la cosa y el mensaje se publica en el tema `$aws/things/My_IoT_Thing/shadow/update/accepted`. Si publica en el tema `$aws/things/My_IoT_Thing/shadow/get`, devuelve una respuesta a la `$aws/things/My_IoT_Thing/shadow/get/accepted` con el estado del dispositivo.

Para las sombras con nombre, primero debe crear la sombra con nombre o publicar una actualización con el nombre de la sombra antes de utilizar la solicitud `get`. Por ejemplo, para crear una sombra con nombre `namedShadow1`, publique primero la información del estado del dispositivo en el tema `$aws/things/My_IoT_Thing/shadow/namedShadow1/update`. Para recuperar la información de estado, utilice el `/get` de la sombra con nombre, `$aws/things/My_IoT_Thing/shadow/namedShadow1/get`.

Procesamiento de mensajes mientras el dispositivo está conectado a AWS IoT

Mientras un dispositivo está conectado a AWS IoT, puede recibir `/update/delta` y debe mantener el estado del dispositivo coincidente con los cambios en sus sombras mediante:

1. Leer todos los mensajes `/update/delta` recibidos y sincronizar el estado del dispositivo para que coincida.
2. Publicando un mensaje `/update` con un cuerpo de mensaje `reported` que tenga el estado actual del dispositivo, siempre que cambie el estado del dispositivo.

Mientras un dispositivo está conectado, debe publicar estos mensajes cuando se indique.

Indicación	Tema	Carga
El estado del dispositivo ha cambiado.	<code>ShadowTopicPrefix/update</code>	Un documento de sombra con la propiedad <code>reported</code> .
Es posible que el dispositivo no esté sincronizado con la sombra.	<code>ShadowTopicPrefix/get</code>	(empty)
Una acción en el dispositivo indica que el dispositivo ya no admite una sombra, por ejemplo, cuando se quita o reemplaza el dispositivo.	<code>ShadowTopicPrefix/delete</code>	(empty)

Procesamiento de mensajes cuando el dispositivo se vuelve a conectar a AWS IoT

Cuando un dispositivo con una o más sombras se conecta a AWS IoT, debe sincronizar su estado con el de todas las sombras que admite del siguiente modo:

1. Leer todos los mensajes /update/delta recibidos y sincronizar el estado del dispositivo para que coincida.
2. Publicando un mensaje /update con un cuerpo de mensaje `reported` que tenga el estado actual del dispositivo.

Uso de sombras en aplicaciones y servicios

En esta sección se describe cómo interactúa una aplicación o un servicio con el servicio Device Shadow de AWS IoT. En este ejemplo se supone que la aplicación o el servicio solo interactúan con la sombra y, a través de la sombra, con el dispositivo. Este ejemplo no incluye ninguna acción de administración, como la creación o eliminación de sombras.

En este ejemplo se utiliza la API REST del servicio Device Shadow de AWS IoT para interactuar con las sombras. A diferencia del ejemplo utilizado en [Uso de sombras en dispositivos \(p. 631\)](#), que utiliza un modelo de comunicaciones de publicación/suscripción, este ejemplo utiliza el modelo de comunicaciones de solicitud/respuesta de la API REST. Esto significa que la aplicación o el servicio deben realizar una solicitud antes de poder recibir una respuesta de AWS IoT. Una desventaja de este modelo, sin embargo, es que no admite notificaciones. Si su aplicación o servicio requiere notificaciones oportunas de cambios en el estado del dispositivo, tenga en cuenta los protocolos MQTT o MQTT sobre WSS, que admiten el modelo de comunicación de publicación/suscripción, tal como se describe en [Uso de sombras en dispositivos \(p. 631\)](#).

Important

Asegúrese de que el uso de las sombras por parte de la aplicación o servicio sea coherente y compatible con las implementaciones correspondientes en los dispositivos. Tenga en cuenta, por ejemplo, cómo se crean, actualizan y eliminan las sombras, y cómo se tratan las actualizaciones en el dispositivo y en las aplicaciones o servicios que acceden a la sombra. El diseño debe especificar claramente cómo se actualiza y notifica el estado del dispositivo, y cómo interactúan las aplicaciones y los servicios con el dispositivo y sus sombras.

La URL de la API REST para sombras con nombre es:

```
https://endpoint/things/thingName/shadow?name=shadowName
```

y para una sombra sin nombre:

```
https://endpoint/things/thingName/shadow
```

donde:

punto de conexión

El punto de enlace que devuelve el comando de la CLI:

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

thingName

El nombre del objeto al que pertenece la sombra

shadowName

El nombre de la sombra con nombre. Este parámetro no se utiliza con sombras sin nombre.

Inicialización de la aplicación o el servicio al conectar a AWS IoT

Cuando la aplicación se conecta por primera vez a AWS IoT, debe enviar una solicitud HTTP GET a las URL de las sombras que utiliza para obtener el estado actual de las sombras que está utilizando. Esto le permite sincronizar la aplicación o el servicio con la sombra.

Procesamiento de cambios de estado mientras la aplicación o el servicio está conectado a AWS IoT

Mientras la aplicación o el servicio está conectado a AWS IoT, puede consultar el estado actual periódicamente enviando una solicitud HTTP GET en las URL de las sombras que utiliza.

Cuando un usuario final interactúa con la aplicación o el servicio para cambiar el estado del dispositivo, la aplicación o el servicio pueden enviar una solicitud HTTP POST a las URL de las sombras que utiliza para actualizar el estado desired de la sombra. Esta solicitud devuelve el cambio que se aceptó, pero es posible que tenga que sondear la sombra realizando solicitudes HTTP GET hasta que el dispositivo haya actualizado la sombra con su nuevo estado.

Detección de un dispositivo conectado

Para determinar si un dispositivo está conectado actualmente, incluya una propiedad connected en el documento de sombra y use un mensaje MQTT Last Will and Testament (LWT) para establecer la propiedad connected en false si un dispositivo está desconectado debido a un error.

Note

Los mensajes MQTT LWT enviados a temas reservados de AWS IoT (temas que comienzan por \$) son omitidos por el servicio Device Shadow de AWS IoT. Sin embargo, los procesan los clientes suscritos y el motor de reglas de AWS IoT, por lo que deberá crear un mensaje LWT que se envíe a un tema no reservado y una regla que vuelva a publicar el mensaje MQTT LWT como un mensaje de actualización de sombra en el tema de actualización reservado de la sombra *ShadowTopicPrefix*/update.

Para enviar al servicio Device Shadow un mensaje LWT

1. Cree una regla que vuelva a publicar el mensaje MQTT LWT en el tema reservado. El siguiente ejemplo es una regla que escucha mensajes en *lamy/things/myLightBulb/update* tema y lo vuelve a publicar en *aws/things/myLightBulb/shadow/update*.

```
{  
  "rule": {  
    "ruleDisabled": false,  
    "sql": "SELECT * FROM 'my/things/myLightBulb/update'",  
    "description": "Turn my/things/ into aws/things/",  
    "actions": [  
      {"topic": "aws/things/myLightBulb/shadow/update", "payload": "ON"}  
    ]  
  }  
}
```

```
{  
    "republish": [  
        {"topic": "$aws/things/myLightBulb/shadow/update",  
         "roleArn": "arn:aws:iam:123456789012:role/aws_iot_republish"}  
    ]  
}
```

2. Cuando el dispositivo se conecta a AWS IoT, registra un mensaje LWT en un tema no reservado para que la regla de republicación lo reconozca. En este ejemplo, ese tema es `$aws/things/myLightBulb/update` y establece la propiedad conectada en `false`.

```
{  
    "state": {  
        "reported": {  
            "connected": "false"  
        }  
    }  
}
```

3. Despues de conectarse, el dispositivo publica un mensaje en su tema de actualización de sombra `$aws/things/myLightBulb/shadow/update`, para notificar su estado actual, que incluye establecer su propiedad `connected` en `true`.

```
{  
    "state": {  
        "reported": {  
            "connected": "true"  
        }  
    }  
}
```

4. Antes de que el dispositivo se desconecte correctamente, publica un mensaje en su tema de actualización de sombras `$aws/things/myLightBulb/shadow/update`, para notificar su estado más reciente, que incluye establecer su propiedad `connected` en `false`.

```
{  
    "state": {  
        "reported": {  
            "connected": "false"  
        }  
    }  
}
```

5. Si el dispositivo se desconecta debido a un error, el agente de mensajes de AWS IoT publica el mensaje LWT del dispositivo en nombre del dispositivo. La regla de republicación detecta este mensaje y publica el mensaje de actualización de sombra para actualizar la propiedad `connected` de la sombra del dispositivo.

Simulación de comunicaciones del servicio Device Shadow

En este tema se muestra cómo el servicio Device Shadow actúa como intermediario y permite que los dispositivos y aplicaciones utilicen una sombra para actualizar, almacenar y recuperar el estado de un dispositivo.

Para demostrar la interacción descrita en este tema y explorarla más a fondo, necesitará una cuenta de AWS y un sistema en el que puede ejecutar la AWS CLI. Si no dispone de ello, aún puede ver la interacción en los ejemplos de código.

En este ejemplo, la consola de AWS IoT representa el dispositivo. La AWS CLI representa la aplicación o servicio que accede al dispositivo a través de la sombra. La interfaz de AWS CLI es muy similar a la API con la que una aplicación podría usar para comunicarse con AWS IoT. El dispositivo de este ejemplo es una bombilla inteligente y la aplicación muestra el estado de la bombilla y puede cambiar el estado de la bombilla.

Configuración de la simulación

Estos procedimientos inicializan la simulación abriendo la [consola de AWS IoT](#), que simula su dispositivo, y la ventana de línea de comandos que simula su aplicación.

Para configurar el entorno de simulación

1. Creación de una cuenta de AWS, si ya tiene uno que usar para esta simulación, puede omitir este paso.

Necesitará una cuenta de AWS para ejecutar los ejemplos de este tema por su cuenta. Si no dispone de una cuenta de AWS, cree uno, como se describe en [Configurar su cuenta de AWS \(p. 18\)](#).

2. Abra la [consola de AWS IoT](#) y, en el menú de la izquierda, elija Probar para abrir el cliente MQTT.
3. En otra ventana, abra una ventana de terminal en un sistema que tenga instalada la AWS CLI.

Debe tener dos ventanas abiertas: una con la consola de AWS IoT en la página Prueba y otra con un símbolo de línea de comandos.

Inicializar el dispositivo

En esta simulación, trabajaremos con un objeto denominado `mySimulatedThing` y su sombra denominada `simShadow1`.

Crear objeto de cosa y su política de IoT

Para crear un objeto de cosa, en la AWS IoT Consola:

1. Elija `Manejar` y luego elija `Objetos`.
2. Haga clic en el botón `.Crear`. Las cosas aparecen en la lista, haga clic en `Registre un solo objeto >crear un solo AWS IoT objeto`.
3. Introduzca el nombre `mySimulatedThing`, deje otros ajustes por defecto y, a continuación, haga clic en `Próximo`.
4. Utilice la creación de certificados con un solo clic para generar los certificados que autenticarán la conexión del dispositivo a AWS IoT. Clic `Activar` para activar el certificado.
5. Puede adjuntar la política `My_IoT_Policy` que permitiría al dispositivo publicar y suscribirse a los temas reservados de MQTT. Para obtener pasos más detallados sobre cómo crear un AWS IoT cosa y cómo crear esta política, consulte [Crear objeto de cosa \(p. 40\)](#).

Crear sombra con nombre para el objeto de cosa

Puede crear una sombra con nombre para una cosa publicando una solicitud de actualización en el tema `$aws/things/mySimulatedThing/shadow/name/simShadow1/update` como se describe a continuación.

Alternativamente, para crear una sombra con nombre:

1. En el navegadorAWS IoTConsola, elige tu objeto de cosa en la lista de cosas que se muestra y, a continuación, eligeSombras.
2. ElegirAñadir una sombra, escriba el nombre`simShadow1`y luego seleccioneCrearpara agregar la sombra con nombre.

Suscribirse y publicar en temas MQTT reservados

En la consola, suscríbase a los temas de sombra reservados de MQTT. Estos temas son las respuestas a las acciones `get`, `update` y `delete` para que el dispositivo esté listo para recibir las respuestas después de publicar una acción.

Para suscribirse a un tema MQTT en el cliente MQTT

1. En el navegadorCliente MQTT, eligeSuscripción a un tema.
2. Introduzca `aget`, `update`, `y delete`temas a los que suscribirse. Copie un tema a la vez de la lista siguiente y péguelo en elFiltro de temasy, a continuación, haga clic enSuscribirse. Debería ver que los temas aparecen enSuscripciones.

- `$aws/things/mySimulatedThing/shadow/name/simShadow1/delete/accepted`
- `$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`

En este punto, el dispositivo simulado está listo para recibir los temas a medida que los publica AWS IoT.

Para publicar en un tema MQTT en laCliente MQTT

Después de que un dispositivo se haya inicializado y suscrito a los temas de respuesta, debe consultar las sombras que admite. Esta simulación solo admite una sombra, la sombra que admite un objeto denominado, `mySimulatedThing`, denominado, `simShadow1`.

Para obtener el estado actual de la sombra desde el Cliente MQTT

1. En el cliente MQTT, elija Publicar en un tema.
2. EnPublicación, introduzca el siguiente tema y elimine cualquier contenido de la ventana del cuerpo del mensaje debajo donde ingresó el tema que desea obtener. A continuación, puede elegirPublicar en temapara publicar la solicitud.`$aws/things/mySimulatedThing/shadow/name/simShadow1/get`.

Si no ha creado la sombra con nombre,`simShadow1`, recibe un mensaje en el`$aws/things/mySimulatedThing/shadow/name/simShadow1/get/rejected`tema y elcodees404, como en este ejemplo, porque la sombra no se ha creado, por lo que la crearemos a continuación.

```
{  
  "code": 404,  
  "message": "No shadow exists with name: 'simShadow1'"  
}
```

Para crear una sombra con el estado actual del dispositivo

1. En el navegadorCliente MQTT, eligePublicación de un tema e introduzca este tema:

```
$aws/things/mySimulatedThing/shadow/name/simShadow1/update
```

2. En la ventana del cuerpo del mensaje debajo donde introdujo el tema, escriba este documento de sombra para mostrar que el dispositivo está notificando su ID y su color actual en valores RGB. ElegirPublicación para publicar la solicitud.

```
{  
    "state": {  
        "reported": {  
            "ID": "SmartLamp21",  
            "ColorRGB": [  
                128,  
                128,  
                128  
            ]  
        }  
    },  
    "clientToken": "426bfd96-e720-46d3-95cd-014e3ef12bb6"  
}
```

Si recibe un mensaje en el tema:

- \$aws/things/mySimulatedThing/shadow/name/simShadow1/update/accepted: significa que se creó la sombra y el cuerpo del mensaje contiene el documento de sombra actual.
- \$aws/things/mySimulatedThing/shadow/name/simShadow1/update/rejected: Revise el error en el cuerpo del mensaje.
- \$aws/things/mySimulatedThing/shadow/name/simShadow1/get/accepted: La sombra ya existe y el cuerpo del mensaje tiene el estado de sombra actual, como en este ejemplo. Con esto, puedes configurar su dispositivo o confirmar que coincide con el estado 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": 1591140517,
    "clientToken": "426bfd96-e720-46d3-95cd-014e3ef12bb6"
}
```

Enviar una actualización desde la aplicación

Esta sección utiliza la AWS CLI para mostrar cómo puede interactuar una aplicación con una sombra.

Para obtener el estado actual de la sombra usando la AWS CLI

Desde la línea de comandos, escriba este comando:

```
aws iot-data get-thing-shadow --thing-name mySimulatedThing --shadow-name simShadow1 /dev/stdout
```

En las plataformas Windows, puede usar conen lugar de /dev/stdout.

```
aws iot-data get-thing-shadow --thing-name mySimulatedThing --shadow-name simShadow1 con
```

Dado que la sombra existe y la ha inicializado el dispositivo para reflejar su estado actual, debe devolver el siguiente 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
}
```

La aplicación puede usar esta respuesta para inicializar su representación del estado del dispositivo.

Si la aplicación actualiza el estado, como cuando un usuario final cambia el color de nuestra bombilla inteligente a amarillo, la aplicación enviará un comando update-thing-shadow. Este comando corresponde a la API REST UpdateThingShadow.

Para actualizar una sombra desde una aplicación

Desde la línea de comandos, escriba 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
```

Si tiene éxito, este comando debe devolver el siguiente 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 a la actualización en el dispositivo

Volviendo al Cliente MQTT en la consola de AWS, debería ver los mensajes que publicó AWS IoT para reflejar el comando de actualización emitido en la sección anterior.

Para ver los mensajes de actualización en el Cliente MQTT

En el Cliente MQTT, elija \$aws/things/mySimulatedThing/shadow/name/simShadow1/update/delta en la columna Suscripciones . Si el nombre del tema está truncado, puede ponerlo en pausa para ver el tema completo. En el registro de temas de este tema, debería ver un /deltaUn mensaje similar al de este.

```
{  
    "version": 4,  
    "timestamp": 1591141596,  
    "state": {  
        "ColorRGB": [  
            255,  
            255,  
            0  
        ]  
    },  
    "metadata": {  
        "ColorRGB": [  
            {  
                "timestamp": 1591141596  
            },  
            {  
                "timestamp": 1591141596  
            },  
            {  
                "timestamp": 1591141596  
            }  
        ]  
    },  
    "clientToken": "21b21b21-bfd2-4279-8c65-e2f697ff4fab"  
}
```

El dispositivo procesaría el contenido de este mensaje para establecer el estado del dispositivo para que coincida con el estado `desired` del mensaje.

Después de que el dispositivo actualice el estado para que coincida con el estado `desired` del mensaje, debe enviar el nuevo estado notificado de nuevo a AWS IoT publicando un mensaje de actualización. Este procedimiento simula esto en el cliente MQTT.

Para actualizar la sombra desde el dispositivo

1. En el cliente MQTT, elija Publicar en un tema.
2. En la ventana del cuerpo del mensaje, en el campo de tema situado encima de la ventana del cuerpo del mensaje, introduzca el tema de la sombra seguido de la /updateacción:\$aws/things/mySimulatedThing/shadow/name/simShadow1/updatey en el cuerpo del mensaje, introduzca este documento de sombra actualizado, que describe el estado actual del dispositivo. ClicPublicaciónpara publicar el estado actualizado del dispositivo.

```
{  
    "state": {  
        "reported": {  
            "ColorRGB": [255,255,0]  
        }  
    },  
    "clientToken": "a4dc2227-9213-4c6a-a6a5-053304f60258"  
}
```

Si el mensaje fue recibido correctamente por AWS IoT, debería ver una nueva respuesta en el registro de mensajes \$aws/things/mySimulatedThing/shadow/name/simShadow1/update/accepted en el Cliente MQTT con el estado actual de la sombra, como este ejemplo.

```
{  
    "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"  
}
```

Una actualización correcta del estado notificado del dispositivo también hace que AWS IoT envíe una descripción completa del estado de sombra en un mensaje al tema , como este cuerpo del mensaje resultante de la actualización instantánea realizada por el dispositivo en el procedimiento anterior.

```
{  
    "previous": {  
        "state": {  
            "desired": {  
                "ColorRGB": [  
                    255,  
                    255,  
                    0  
                ]  
            }  
        },  
        "reported": {  
            "ID": "SmartLamp21",  
            "ColorRGB": [  
                128,  
                128,  
                128  
            ]  
        }  
    },  
    "metadata": {  
        "desired": {  
            "ColorRGB": [  
                {  
                    "timestamp": 1591141596  
                },  
                {  
                    "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"
}
```

Observe la actualización en la aplicación

La aplicación ahora puede consultar la sombra del estado actual según haya notificado el dispositivo.

Para obtener el estado actual de la sombra usando la AWS CLI

1. Desde la línea de comandos, escriba este comando:

```
aws iot-data get-thing-shadow --thing-name mySimulatedThing --shadow-name simShadow1 /  
dev/stdout
```

En las plataformas Windows, puede usar conen lugar de /dev/stdout.

```
aws iot-data get-thing-shadow --thing-name mySimulatedThing --shadow-name simShadow1  
con
```

2. Dado que el dispositivo acaba de actualizar la sombra para reflejar su estado actual, debe devolver el siguiente 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
}
```

Más allá de la simulación

Experimente con la interacción entre el AWS CLI (que representa la aplicación) y la consola (que representa el dispositivo) para modelar su solución IoT.

Interacción con sombras

En este tema se describen los mensajes asociados a cada uno de los tres métodos que proporciona AWS IoT para trabajar con sombras. Estos métodos incluyen lo siguiente:

UPDATE

Crea una sombra si no existe o actualiza el contenido de una sombra existente con la información de estado proporcionada en el cuerpo del mensaje. AWS IoT registra una marca temporal con cada actualización para indicar cuándo se actualizó el estado por última vez. Cuando cambia el estado de la sombra, AWS IoT envía mensajes `/delta` a todos los suscriptores de MQTT con la diferencia entre los estados `desired` y `reported`. Los dispositivos o aplicaciones que reciben un mensaje `/delta` pueden realizar acciones en función de la diferencia. Por ejemplo, un dispositivo puede actualizar su estado al estado deseado o una aplicación puede actualizar su interfaz de usuario para mostrar el cambio de estado del dispositivo.

GET

Recupera un documento de sombra actual que contiene el estado completo de la sombra, incluidos los metadatos.

DELETE

Elimina la sombra del dispositivo y su contenido.

No se puede restaurar un documento de sombra de dispositivo eliminado, pero se puede crear otra con el nombre de un documento de sombra de dispositivo eliminado. Si crea un documento de sombra de dispositivo que tiene el mismo nombre que uno eliminado en las últimas 48 horas, el número de versión del nuevo documento sombra de dispositivo seguirá al del eliminado. Si se ha eliminado un

documento de sombra de dispositivo durante más de 48 horas, el número de versión de un nuevo documento sombra de dispositivo con el mismo nombre será 0.

Compatibilidad del protocolo

AWS IoT admite [MQTT](#) y una API REST a través de protocolos HTTPS para interactuar con sombras. AWS IoT proporciona un conjunto de temas de solicitud y respuesta reservados para acciones de publicación y suscripción de MQTT. Los dispositivos y las aplicaciones deben suscribirse a los temas de respuesta antes de publicar un tema de solicitud para obtener información sobre cómo trató AWS IoT la solicitud. Para obtener más información, consulte [Temas MQTT de sombra de dispositivo \(p. 657\)](#) y [API REST de sombra de dispositivo \(p. 653\)](#).

Estado de solicitud y notificación

Al diseñar su solución IoT usando AWS IoT y sombras, debe determinar las aplicaciones o dispositivos que solicitarán cambios y los que los implementarán. Normalmente, un dispositivo implementa y notifica los cambios a la sombra y las aplicaciones y los servicios responden y solicitan cambios en la sombra. Su solución podría ser diferente, pero en los ejemplos de este tema se supone que la aplicación cliente o el servicio solicita cambios en la sombra y el dispositivo realiza los cambios y los notifica de nuevo a la sombra.

Actualización de la sombra

La aplicación o el servicio pueden actualizar el estado de una sombra mediante la API [UpdateThingShadow \(p. 654\)](#) o publicando en el tema [/update \(p. 660\)](#). Las actualizaciones afectan únicamente a los campos especificados en la solicitud.

Actualización de una sombra cuando un cliente solicita un cambio de estado

Cuando un cliente solicita un cambio de estado en una sombra mediante el protocolo MQTT

1. El cliente debe tener un documento de sombra actual para que pueda identificar las propiedades que se van a cambiar. Consulte la acción `/get` para ver cómo obtener el documento de sombra actual.
2. El cliente se suscribe a los siguientes temas de 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. El cliente publica un tema de solicitud de `$aws/things/thingName/shadow/name/shadowName/update` con un documento de estado que contiene el estado deseado de la sombra. Solo las propiedades que se van a cambiar deben incluirse en el documento. Este es un ejemplo de un documento con el estado deseado.

```
{  
  "state": {  
    "desired": {  
      "color": {  
        "r": 10  
      },  
      "engine": "ON"  
    }  
  }  
}
```

```
}
```

4. Si la solicitud de actualización es válida, AWS IoT actualiza el estado deseado en la sombra y publica mensajes sobre estos temas:
 - \$aws/things/*thingName*/shadow/name/*shadowName*/update/accepted
 - \$aws/things/*thingName*/shadow/name/*shadowName*/update/delta

El mensaje /update/accepted contiene un documento de sombra [Documento de estado de la respuesta /aceptado \(p. 666\)](#) y el mensaje /update/delta contiene un documento de sombra [Documento de estado de la respuesta /delta \(p. 667\)](#).

5. Si la solicitud de actualización no es válida, AWS IoT publica un mensaje con el tema \$aws/things/*thingName*/shadow/name/*shadowName*/update/rejected con un documento de sombra [Documento de respuesta de error \(p. 669\)](#) que describe el error.

Cuando un cliente solicita un cambio de estado en una sombra mediante el uso de la API

1. El cliente llama a la API [UpdateThingShadow \(p. 654\)](#) con un documento de estado [Documento de estado de la solicitud \(p. 666\)](#) como cuerpo del mensaje.
2. Si la solicitud fuera válida, AWS IoT devuelve un código de respuesta HTTP correcta y un documento de sombra [Documento de estado de la respuesta /aceptado \(p. 666\)](#) como cuerpo del mensaje de respuesta.

AWS IoT también publicará un mensaje MQTT en el tema \$aws/things/*thingName*/shadow/name/*shadowName*/update/delta con un documento de sombra [Documento de estado de la respuesta /delta \(p. 667\)](#) para cualquier dispositivo o cliente que se suscriba a él.

3. Si la solicitud no era válida, AWS IoT devuelve un código de respuesta de error HTTP y [Documento de respuesta de error \(p. 669\)](#) como cuerpo del mensaje de respuesta.

Cuando el dispositivo recibe el estado /desired en el tema /update/delta, realiza los cambios deseados en el dispositivo. A continuación, envía un mensaje al tema /update para notificar su estado actual a la sombra.

Actualización de una sombra cuando un dispositivo notifica su estado actual

Cuando un dispositivo notifica su estado actual a la sombra mediante el protocolo MQTT

1. El dispositivo debe suscribirse a estos temas de MQTT antes de actualizar la 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. El dispositivo notifica su estado actual publicando un mensaje en el tema \$aws/things/*thingName*/shadow/name/*shadowName*/update que notifica estado actual, como en este ejemplo.

```
{
    "state": {
        "reported" : {
            "color" : { "r" : 10 },
            "engine" : "ON"
        }
    }
}
```

```
        }
    }
```

3. Si AWS IoT acepta la actualización, publica un mensaje en los temas \$aws/things/*thingName*/shadow/name/*shadowName*/update/accepted con un documento de sombra [Documento de estado de la respuesta /aceptado \(p. 666\)](#).
4. Si la solicitud de actualización no es válida, AWS IoT publica un mensaje con el tema \$aws/things/*thingName*/shadow/name/*shadowName*/update/rejected con un documento de sombra [Documento de respuesta de error \(p. 669\)](#) que describe el error.

Cuando un dispositivo notifica su estado actual a la sombra mediante la API

1. El dispositivo llama a la API [UpdateThingShadow \(p. 654\)](#) con un documento de estado [Documento de estado de la solicitud \(p. 666\)](#) como cuerpo del mensaje.
2. Si la solicitud era válida, AWS IoT actualiza la sombra y devuelve un código de respuesta HTTP correcto con un documento de sombra [Documento de estado de la respuesta /aceptado \(p. 666\)](#) como cuerpo del mensaje de respuesta.

AWS IoT también publicará un mensaje MQTT en el tema \$aws/things/*thingName*/shadow/name/*shadowName*/update/delta con un documento de sombra [Documento de estado de la respuesta /delta \(p. 667\)](#) para cualquier dispositivo o cliente que se suscriba a él.

3. Si la solicitud no era válida, AWS IoT devuelve un código de respuesta de error HTTP y [Documento de respuesta de error \(p. 669\)](#) como cuerpo del mensaje de respuesta.

Bloqueo optimista

Puede utilizar la versión del documento de estado para asegurarse de que actualiza la versión más reciente del documento de sombra de un dispositivo. Cuando se suministra una versión con una solicitud de actualización, el servicio rechaza la solicitud con un código de respuesta de conflicto HTTP 409 si la versión actual del documento de estado no coincide con la versión suministrada.

Por ejemplo:

Documento inicial:

```
{
  "state": {
    "desired": {
      "colors": [
        "RED",
        "GREEN",
        "BLUE"
      ]
    }
  },
  "version": 10
}
```

Actualización: (la versión no coincide; se rechazará esta solicitud)

```
{
  "state": {
    "desired": {
      "colors": [
        "BLUE"
      ]
    }
  }
}
```

```
    },
    "version": 9
}
```

Resultado:

```
{
  "code": 409,
  "message": "Version conflict",
  "clientToken": "426bfd96-e720-46d3-95cd-014e3ef12bb6"
}
```

Actualización: (la versión coincide; esta solicitud se aceptará)

```
{
  "state": {
    "desired": {
      "colors": [
        "BLUE"
      ]
    }
  },
  "version": 10
}
```

Estado final:

```
{
  "state": {
    "desired": {
      "colors": [
        "BLUE"
      ]
    }
  },
  "version": 11
}
```

Recuperación de un documento de sombra

Puede recuperar un documento de sombra utilizando la API [GetThingShadow \(p. 654\)](#) o suscribiéndose y publicando en el tema [/get \(p. 658\)](#). Esto recupera un documento de sombra completo, incluido cualquier delta entre los estados `desired` y `reported`. El procedimiento para esta tarea es el mismo si el dispositivo o un cliente está realizando la solicitud.

Para recuperar un documento de sombra mediante el protocolo MQTT

1. El dispositivo o cliente debe suscribirse a estos temas de MQTT antes de actualizar la sombra:
 - `$aws/things/thingName/shadow/name/shadowName/get/accepted`
 - `$aws/things/thingName/shadow/name/shadowName/get/rejected`
2. El dispositivo o cliente publica un mensaje en el tema `$aws/things/thingName/shadow/name/shadowName/get` con un cuerpo de mensaje vacío.
3. Si la solicitud se realiza correctamente, AWS IoT publica un mensaje en el tema `$aws/things/thingName/shadow/name/shadowName/get/accepted` con un Documento de estado de la respuesta [aceptado \(p. 666\)](#) en el cuerpo del mensaje.

4. Si la solicitud no era válida, AWS IoT publica un mensaje en el tema `$aws/things/thingName/shadow/name/shadowName/get/rejected` con un [Documento de respuesta de error \(p. 669\)](#) en el cuerpo del mensaje.

Para recuperar un documento de sombra mediante una API REST

1. El dispositivo o cliente llama a la API [GetThingShadow \(p. 654\)](#) con un cuerpo de mensaje vacío.
2. Si la solicitud es válida, AWS IoT devuelve un código de respuesta HTTP correcto con un documento de sombra [Documento de estado de la respuesta /aceptado \(p. 666\)](#) como cuerpo del mensaje de respuesta.
3. Si la solicitud no es válida, AWS IoT devuelve un código de respuesta de error HTTP y [Documento de respuesta de error \(p. 669\)](#) como cuerpo del mensaje de respuesta.

Eliminación de datos de sombra

Hay dos formas de eliminar datos de sombra: puede eliminar propiedades específicas en el documento de sombra y puede eliminar la sombra por completo.

- Para eliminar propiedades específicas de una sombra, actualice la sombra; sin embargo, establezca el valor de las propiedades que desea eliminar en `null`. Los campos con un valor `null` se quitan del documento de sombra.
- Para eliminar toda la sombra, use la API [DeleteThingShadow \(p. 655\)](#) o publique en el tema `/delete` (p. 663).

Tenga en cuenta que la eliminación de una sombra no restablece su número de versión a 0.

Eliminación de una propiedad de un documento de sombra

Para eliminar una propiedad de una sombra mediante el protocolo MQTT

1. El dispositivo o cliente debe tener un documento de sombra actual para que pueda identificar las propiedades que se van a cambiar. Consulte [Recuperación de un documento de sombra \(p. 650\)](#) para obtener información sobre cómo obtener el documento de sombra actual.
2. El dispositivo o cliente se suscribe a estos temas de MQTT:
 - `$aws/things/thingName/shadow/name/shadowName/update/accepted`
 - `$aws/things/thingName/shadow/name/shadowName/update/rejected`
3. El dispositivo o cliente publica un tema de solicitud `$aws/things/thingName/shadow/name/shadowName/update` con un documento de estado que asigna valores `null` a las propiedades de la sombra que se va a eliminar. Solo las propiedades que se van a cambiar deben incluirse en el documento. Este es un ejemplo de un documento que elimina la propiedad `engine`.

```
{  
  "state": {  
    "desired": {  
      "engine": null  
    }  
  }  
}
```

4. Si la solicitud de actualización es válida, AWS IoT elimina las propiedades especificadas en la sombra y publica un mensaje con el tema `$aws/things/thingName/shadow/name/shadowName/update/accepted` con un documento de sombra [Documento de estado de la respuesta /aceptado \(p. 666\)](#) en el cuerpo del mensaje.

5. Si la solicitud de actualización no es válida, AWS IoT publica un mensaje con el tema \$aws/things/*thingName*/shadow/name/*shadowName*/update/rejected con un documento de sombra Documento de respuesta de error (p. 669) que describe el error.

Para eliminar una propiedad de una sombra mediante la API REST

1. El dispositivo o cliente llama a la API [UpdateThingShadow \(p. 654\)](#) con un Documento de estado de la solicitud (p. 666) que asigna valores null a las propiedades de la sombra para eliminar. Incluya solo las propiedades que deseé eliminar en el documento. Este es un ejemplo de un documento que elimina la propiedad engine.

```
{  
    "state": {  
        "desired": {  
            "engine": null  
        }  
    }  
}
```

2. Si la solicitud fuera válida, AWS IoT devuelve un código de respuesta HTTP correcta y un documento de sombra Documento de estado de la respuesta /aceptado (p. 666) como cuerpo del mensaje de respuesta.
3. Si la solicitud no era válida, AWS IoT devuelve un código de respuesta de error HTTP y Documento de respuesta de error (p. 669) como cuerpo del mensaje de respuesta.

Eliminación de una sombra

Note

Al establecer el estado de sombra del dispositivo en null no se elimina la sombra. La versión de sombra se incrementará en la próxima actualización.

La eliminación de la sombra de un dispositivo no elimina el objeto. La eliminación de un objeto no elimina la sombra del dispositivo correspondiente.

Eliminar una sombra no restablece su número de versión a 0.

Para eliminar una sombra mediante el protocolo MQTT

1. El dispositivo o cliente se suscribe a estos temas de MQTT:
 - \$aws/things/*thingName*/shadow/name/*shadowName*/delete/accepted
 - \$aws/things/*thingName*/shadow/name/*shadowName*/delete/rejected
2. El dispositivo o cliente publica un \$aws/things/*thingName*/shadow/name/*shadowName*/delete con un búfer de mensaje vacío.
3. Si la solicitud de eliminación es válida, AWS IoT elimina la sombra y publica un mensaje con el tema \$aws/things/*thingName*/shadow/name/*shadowName*/delete/accepted y un documento de sombra Documento de estado de la respuesta /aceptado (p. 666) abreviado en el cuerpo del mensaje. Este es un ejemplo del mensaje de eliminación aceptado:

```
{  
    "version": 4,  
    "timestamp": 1591057529  
}
```

4. Si la solicitud de actualización no es válida, AWS IoT publica un mensaje con el tema \$aws/things/*thingName*/shadow/name/*shadowName*/delete/rejected con un documento de sombra Documento de respuesta de error (p. 669) que describe el error.

Para eliminar una sombra mediante la API REST

1. El dispositivo o cliente llama a la API [DeleteThingShadow](#) (p. 655) con un búfer de mensaje vacío.
2. Si la solicitud era válida, AWS IoT devuelve un código de respuesta HTTP correcta [Documento de estado de la respuesta /aceptado \(p. 666\)](#) y un documento de sombra [Documento de estado de la respuesta /aceptado \(p. 666\)](#) abreviado en el cuerpo del mensaje. Este es un ejemplo del mensaje de eliminación aceptado:

```
{  
    "version": 4,  
    "timestamp": 1591057529  
}
```

3. Si la solicitud no era válida, AWS IoT devuelve un código de respuesta de error HTTP y [Documento de respuesta de error \(p. 669\)](#) como cuerpo del mensaje de respuesta.

API REST de sombra de dispositivo

Una sombra expone el siguiente URI para actualizar la información de estado:

```
https://account-specific-prefix-ats.iot.region.amazonaws.com/things/<thingName>/shadow
```

El punto de enlace específico de suCuenta de AWS. Para buscar el punto de enlace, puede:

- Usar[describe-endpoint](#)desde laAWS CLI.
- UsarAWS IoTconfiguración de consola. EnConfiguración, el punto final aparece en la listaPunto de enlace personalizado
- UsarAWS IoTPágina de detalles de cosa de consola. En la consola de :
 1. AbiertoManejary enManejar, eligeObjetos.
 2. En la lista de cosas, elija la cosa para la que desea obtener el URI del endpoint.
 3. Elija el iconoSombras del dispositivoy elija su sombra. Puede ver el URI de endpoint en elURL Device ShadowSección sobre de laDetalles Device Shadow(Se ha creado el certificado).

El formato del punto de enlace es el siguiente:

```
<identifier>.iot.region.amazonaws.com
```

La API de REST de sombras sigue los mismos mapeos de puertos y protocolos HTTPS que se describen en [Protocolos de comunicación de dispositivos \(p. 81\)](#).

Note

Para usar las API, debe usariotdevicegatewaycomo nombre de servicio para la autenticación. Para obtener más información, consulte[Plan de datos IoT](#).

Acciones de API

- [GetThingShadow](#) (p. 654)
- [UpdateThingShadow](#) (p. 654)
- [DeleteThingShadow](#) (p. 655)
- [ListNamedShadowsForThing](#) (p. 656)

También puede usar la API de para crear una sombra con nombre proporcionandóname=*shadowName* como parte del parámetro de consulta de la API.

GetThingShadow

Obtiene la sombra de objeto especificado.

El documento de estado de respuesta incluye el delta entre los estados `desired` y `reported`.

Solicitud

La solicitud incluye los encabezados HTTP estándar y el URI siguiente:

```
HTTP GET https://endpoint/things/thingName/shadow?name=shadowName
Request body: (none)
```

El parámetro de consulta `name` no es necesario para sombras sin nombre (clásicas).

Respuesta

En caso de éxito, la respuesta incluye encabezados HTTP estándar, así como el código y el cuerpo siguientes:

```
HTTP 200
Response Body: response state document
```

Para obtener más información, consulte [Ejemplo de documento de estado de respuesta \(p. 666\)](#).

Autorización

Para recuperar una sombra, se necesita una política que permita al intermediario ejecutar la acción `iot:GetThingShadow`. El servicio Device Shadow acepta dos formas de autenticación: Signature Version 4 con credenciales de IAM o autenticación mutua TLS con un certificado de cliente.

A continuación, se muestra una política de ejemplo que permite a un intermediario recuperar la sombra de un dispositivo:

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

UpdateThingShadow

Actualiza la sombra del objeto especificado.

Las actualizaciones solo afectan a los campos especificados en el documento de estado de la solicitud. Todos los campos que tengan el valor `null` se eliminarán de la sombra del dispositivo.

Solicitud

La solicitud incluye los encabezados HTTP estándar, así como el URI y el cuerpo siguientes:

```
HTTP POST https://endpoint/things/thingName/shadow?name=shadowName
Request body: request state document
```

El parámetro de consulta `name` no es necesario para sombras sin nombre (clásicas).

Para obtener más información, consulte [Ejemplo de documento de estado de solicitud \(p. 666\)](#).

Respuesta

En caso de éxito, la respuesta incluye encabezados HTTP estándar, así como el código y el cuerpo siguientes:

```
HTTP 200
Response body: response state document
```

Para obtener más información, consulte [Ejemplo de documento de estado de respuesta \(p. 666\)](#).

Autorización

Para actualizar una sombra se necesita una política que permita al intermediario ejecutar la acción `iot:UpdateThingShadow`. El servicio Device Shadow acepta dos formas de autenticación: Signature Version 4 con credenciales de IAM o autenticación mutua TLS con un certificado de cliente.

A continuación, se muestra una política de ejemplo que permite a un intermediario actualizar la sombra de un dispositivo:

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

DeleteThingShadow

Elimina la sombra de objeto especificado.

Solicitud

La solicitud incluye los encabezados HTTP estándar y el URI siguiente:

```
HTTP DELETE https://endpoint/things/thingName/shadow?name=shadowName
Request body: (none)
```

El parámetro de consulta `name` no es necesario para sombras sin nombre (clásicas).

Respuesta

En caso de éxito, la respuesta incluye encabezados HTTP estándar, así como el código y el cuerpo siguientes:

```
HTTP 200
Response body: Empty response state document
```

Tenga en cuenta que la eliminación de una sombra no restablece su número de versión a 0.

Autorización

Para eliminar la sombra de un dispositivo se necesita una política que permita al intermediario ejecutar la acción `iot:DeleteThingShadow`. El servicio Device Shadow acepta dos formas de autenticación: Signature Version 4 con credenciales de IAM o autenticación mutua TLS con un certificado de cliente.

A continuación, se muestra una política de ejemplo que permite a un intermediario eliminar la sombra de un dispositivo:

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

ListNamedShadowsForThing

Muestra las sombras del objeto especificado.

Solicitud

La solicitud incluye los encabezados HTTP estándar y el URI siguiente:

```
HTTP GET /api/things/shadow/ListNamedShadowsForThing/thingName?
nextToken=nextToken&pageSize=pageSize
Request body: (none)
```

nextToken

El token para recuperar el siguiente grupo de resultados.

Este valor se devuelve en los resultados paginados y se utiliza en la llamada que devuelve la página siguiente.

pageSize

El número de nombres de sombra que devolver en cada llamada. Véase también `nextToken`.

thingName

El nombre del objeto para el que mostrar las sombras con nombre.

Respuesta

En caso de éxito, la respuesta incluye encabezados HTTP estándar, así como el código de respuesta siguiente y un [Documento de respuesta de lista de nombres de sombra \(p. 669\)](#).

Note

La sombra sin nombre (clásica) no aparece en esta lista. La respuesta es una lista vacía si solo tienes una sombra clásica o si el `thingName` que especifique no existe.

```
HTTP 200
Response body: Shadow name list document
```

Autorización

Para enumerar la sombra de un dispositivo se necesita una política que permita al intermediario ejecutar `iot:ListNamedShadowsForThing` acción. El servicio Device Shadow acepta dos formas de autenticación: Signature Version 4 con credenciales de IAM o autenticación mutua TLS con un certificado de cliente.

A continuación, se muestra una política de ejemplo que permite a un intermediario mostrar las sombras con nombre de un objeto:

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

Temas MQTT de sombra de dispositivo

El servicio Device Shadow utiliza temas MQTT reservados para permitir a los dispositivos y las aplicaciones obtener, actualizar o eliminar la información de estado de un dispositivo (sombra).

La publicación y suscripción de temas de sombra requiere una autorización basada en temas. AWS IoT se reserva el derecho a añadir nuevos temas a la estructura de temas existente. Por este motivo, le recomendamos que evite las suscripciones de tipo comodín a los temas de sombra. Por ejemplo, evite suscribirse a filtros de temas como `$aws/things/thingName/shadow/#`, ya que el número de temas que coinciden con este filtro de temas puede aumentar a medida que AWS IoT incorpora nuevos temas de sombra. Para consultar ejemplos de mensajes publicados en estos temas vaya a [Interacción con sombras \(p. 646\)](#).

Las sombras pueden ser con nombre o sin nombre (clásico). Los temas utilizados por cada uno solo difieren en el prefijo del tema. Esta tabla muestra el prefijo de tema utilizado por cada tipo de sombra.

Valor <code>ShadowTopicPrefix</code>	Tipo de sombra
<code>\$aws/things/<i>thingName</i>/shadow</code>	Sombra sin nombre (clásica)
<code>\$aws/things/<i>thingName</i>/shadow/<i>name</i>/<i>shadowName</i></code>	Sombra con nombre

Para crear un tema completo, seleccione el *ShadowTopicPrefix* para el tipo de sombra al que desea hacer referencia, reemplace *thingName* y *shadowName* si procede, por sus valores correspondientes y, a continuación, anéxelo al código auxiliar del tema como se muestra en las secciones siguientes.

A continuación se muestran los temas MQTT utilizados para interactuar con las sombras.

Temas

- [/get \(p. 658\)](#)
- [/get/accepted \(p. 658\)](#)
- [/get/rejected \(p. 659\)](#)
- [/update \(p. 660\)](#)
- [/update/delta \(p. 661\)](#)
- [/update/accepted \(p. 661\)](#)
- [/update/documents \(p. 662\)](#)
- [/update/rejected \(p. 663\)](#)
- [/delete \(p. 663\)](#)
- [/delete/accepted \(p. 664\)](#)
- [/delete/rejected \(p. 665\)](#)

/get

Publique un mensaje vacío en este tema para obtener la sombra de objeto:

```
ShadowTopicPrefix/get
```

AWS IoT responde publicando en [/get/accepted \(p. 658\)](#) o en [/get/rejected \(p. 659\)](#).

Política de ejemplo

A continuación, mostramos un ejemplo de la política requerida:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:Publish"
            ],
            "Resource": [
                "arn:aws:iot:region:account:topic/$aws/things/thingName/shadow/get"
            ]
        }
    ]
}
```

/get/accepted

AWS IoT publica un documento de sombra de respuesta en este tema cuando devuelve la sombra del dispositivo:

```
ShadowTopicPrefix/get/accepted
```

Para obtener más información, consulte [Documentos de estado de la respuesta \(p. 666\)](#).

Política de ejemplo

A continuación, mostramos un ejemplo de la política requerida:

```
{  
    "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

AWS IoT publica un documento de respuesta de error en este tema cuando no puede devolver la sombra del dispositivo:

```
ShadowTopicPrefix/get/rejected
```

Para obtener más información, consulte [Documento de respuesta de error \(p. 669\)](#).

Política de ejemplo

A continuación, mostramos un ejemplo de la política requerida:

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

```
        "arn:aws:iot:region:account:topic/$aws/things/thingName/shadow/get/rejected"
    }
}
```

/update

Publique un documento de estado de solicitud en este tema para actualizar la sombra del dispositivo:

```
ShadowTopicPrefix/update
```

El cuerpo del mensaje contiene un [documento de estado de solicitud parcial \(p. 666\)](#).

Un cliente que intente actualizar el estado de un dispositivo enviaría un documento de estado de solicitud JSON con la propiedad `desired` como esta:

```
{
  "state": {
    "desired": {
      "color": "red",
      "power": "on"
    }
  }
}
```

Un dispositivo que actualice su sombra enviaría un documento de estado de solicitud JSON con la propiedad `reported`, como esta:

```
{
  "state": {
    "reported": {
      "color": "red",
      "power": "on"
    }
  }
}
```

AWS IoT responde publicando en [/update/accepted \(p. 661\)](#) o en [/update/rejected \(p. 663\)](#).

Política de ejemplo

A continuación, mostramos un ejemplo de la política requerida:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iot:Publish"
      ],
      "Resource": [
        "arn:aws:iot:region:account:topic/$aws/things/thingName/shadow/update"
      ]
    }
  ]
}
```

/update/delta

AWS IoT publica un documento de estado de respuesta en este tema cuando acepta un cambio de la sombra del dispositivo y el documento de estado de solicitud contiene valores distintos para los estados `desired` y `reported`:

```
ShadowTopicPrefix/update/delta
```

El búfer del mensaje contiene un Documento de estado de la respuesta /delta (p. 667).

Detalles del cuerpo del mensaje

- Un mensaje publicado en `update/delta` incluye únicamente los atributos deseados que difieren entre las secciones `desired` y `reported`. Contiene todos estos atributos, independientemente de si se encuentran en el mensaje de actualización actual o si ya estaban almacenados en AWS IoT. No se incluyen los atributos que no difieren entre las secciones `desired` y `reported`.
- Si un atributo se encuentra en la sección `reported`, pero no tiene equivalente en la sección `desired`, no se incluye.
- Si un atributo se encuentra en la sección `desired`, pero no tiene equivalente en la sección `reported`, se incluye.
- Si se elimina un atributo de la sección `reported`, pero sigue existiendo en la sección `desired`, se incluye.

Política de ejemplo

A continuación, mostramos un ejemplo de la política requerida:

```
{
  "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

AWS IoT publica un documento de estado de respuesta en este tema cuando acepta un cambio de la sombra del dispositivo:

ShadowTopicPrefix/update/accepted

El búfer del mensaje contiene un [Documento de estado de la respuesta /aceptado](#) (p. 666).

Política de ejemplo

A continuación, mostramos un ejemplo de la política requerida:

```
{  
    "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

AWS IoT publica un documento de estado en este tema siempre que se realiza una actualización correcta de la sombra:

ShadowTopicPrefix/update/documents

El cuerpo del mensaje contiene un [Documento de estado de respuesta /documentos](#) (p. 667).

Política de ejemplo

A continuación, mostramos un ejemplo de la política requerida:

```
{  
    "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

AWS IoT publica un documento de respuesta de error en este tema cuando rechaza un cambio de la sombra del dispositivo:

```
ShadowTopicPrefix/update/rejected
```

El cuerpo del mensaje contiene un [Documento de respuesta de error \(p. 669\)](#).

Política de ejemplo

A continuación, mostramos un ejemplo de la política requerida:

```
{
    "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 eliminar la sombra de un dispositivo, publique un mensaje vacío para eliminar el tema:

```
ShadowTopicPrefix/delete
```

El contenido del mensaje no se tiene en cuenta.

Tenga en cuenta que la eliminación de una sombra no restablece su número de versión a 0.

AWS IoT responde publicando en [/delete/accepted \(p. 664\)](#) o en [/delete/rejected \(p. 665\)](#).

Política de ejemplo

A continuación, mostramos un ejemplo de la política requerida:

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "iot:Publish"  
            ],  
            "Resource": [  
                "arn:aws:iot:region:account:topic/$aws/things/thingName/shadow/delete"  
            ]  
        }  
    ]  
}
```

/delete/accepted

AWS IoT publica un mensaje en este tema cuando se elimina la sombra de un dispositivo:

```
ShadowTopicPrefix/delete/accepted
```

Política de ejemplo

A continuación, mostramos un ejemplo de la política requerida:

```
{  
    "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

AWS IoT publica un documento de respuesta de error en este tema cuando no puede eliminar la sombra del dispositivo:

```
ShadowTopicPrefix/delete/rejected
```

El cuerpo del mensaje contiene un [Documento de respuesta de error \(p. 669\)](#).

Política de ejemplo

A continuación, mostramos un ejemplo de la política requerida:

```
{
  "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 del servicio Device Shadow

El servicio Device Shadow respeta todas las reglas de la especificación JSON. Los valores, objetos y matrices se almacenan en el documento de sombra del dispositivo.

Contenido

- [Ejemplos de documento de sombra \(p. 665\)](#)
- [Propiedades del documento \(p. 670\)](#)
- [Estado delta \(p. 671\)](#)
- [Control de versiones de documentos de sombra \(p. 672\)](#)
- [Tokens de cliente en documentos de sombra \(p. 672\)](#)
- [Propiedades de documento de sombra vacío \(p. 672\)](#)
- [Valores de matriz en documentos sombra \(p. 673\)](#)

Ejemplos de documento de sombra

El servicio Device Shadow utiliza los siguientes documentos en operaciones UPDATE, GET y DELETE mediante la [API REST \(p. 653\)](#) o [mensajes de publicación/suscripción MQTT \(p. 657\)](#).

Ejemplos

- [Documento de estado de la solicitud \(p. 666\)](#)
- [Documentos de estado de la respuesta \(p. 666\)](#)
- [Documento de respuesta de error \(p. 669\)](#)
- [Documento de respuesta de lista de nombres de sombra \(p. 669\)](#)

Documento de estado de la solicitud

Un documento de estado de solicitud tiene el siguiente formato:

```
{  
    "state": {  
        "desired": {  
            "attribute1": "integer2",  
            "attribute2": "string2",  
            ...  
            "attributeN": "boolean2"  
        },  
        "reported": {  
            "attribute1": "integer1",  
            "attribute2": "string1",  
            ...  
            "attributeN": "boolean1"  
        }  
    },  
    "clientToken": "token",  
    "version": "version"  
}
```

- **state**: las actualizaciones solo afectan a los campos especificados. Normalmente, utilizará la propiedad **desired** o la propiedad **reported**, pero no ambas en la misma solicitud.
- **desired**— Las propiedades y valores de estado solicitados para actualizarse en el dispositivo.
- **reported**— Las propiedades y valores de estado notificados por el dispositivo.
- **clientToken**: si se utiliza, puede hacer coincidir la solicitud y la respuesta correspondiente mediante el token de cliente.
- **version**: si se utiliza, el servicio Device Shadow procesa la actualización solo si la versión especificada coincide con la versión más reciente que tiene.

Documentos de estado de la respuesta

Los documentos de estado de respuesta tienen el siguiente formato en función del tipo de respuesta.

Documento de estado de la respuesta /aceptado

```
{  
    "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 de la respuesta /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 de respuesta /documentos

```
{
    "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"
}
```

Propiedades del documento de estado de la respuesta

- **previous**— Tras una actualización correcta, contiene el state del objeto antes de la actualización.
- **current**— Tras una actualización correcta, contiene el state del objeto tras la actualización.
- **state**
 - **reported**— Presente solo si una cosa informó de algún dato en el reported y contiene únicamente los campos que se encontraban en el documento de estado de la solicitud.
 - **desired**— Presente solo si un dispositivo ha informado de algún dato en el desired y contiene únicamente los campos que se encontraban en el documento de estado de la solicitud.
 - **delta**— Presente solo si el desired los datos difieren de la actual sombra reported data.
- **metadata**: contiene las marcas temporales de cada atributo de las secciones desired y reported para que se pueda determinar cuándo se actualizó el estado.
- **timestamp** fecha y hora de inicio en que AWS IoT generó la respuesta.
- **clientToken** solo está presente si se ha utilizando un token de cliente al publicar un JSON válido en el tema /update.
- **version**: la versión actual del documento de la sombra del dispositivo compartido en AWS IoT. Se aumenta en una unidad con relación a la versión anterior del documento.

Documento de respuesta de error

Un documento de respuesta de error tiene el formato siguiente:

```
{
    "code": error-code,
    "message": error-message",
    "timestamp": timestamp,
    "clientToken": "token"
}
```

- **code** código de respuesta HTTP que indica el tipo de error.
- **message** mensaje de texto que proporciona información adicional.
- **timestamp** fecha y hora en que AWS IoT generó la respuesta. Esta propiedad no está presente en todos los documentos de respuesta de error.
- **clientToken**: solo está presente si se ha utilizando un token de cliente en el mensaje publicado.

Para obtener más información, consulte [Mensajes de error de Device Shadow \(p. 674\)](#).

Documento de respuesta de lista de nombres de sombra

Un documento de respuesta de lista de nombre de sombra tiene el siguiente formato:

```
{
```

```
"results": [
    "shadowName-1",
    "shadowName-2",
    "shadowName-3",
    "shadowName-n"
],
"nextToken": "nextToken",
"timestamp": timestamp
}
```

- **results**— La matriz de nombres de sombras.
- **nextToken**— El valor del token que se utilizará en las solicitudes paginadas para obtener la siguiente página de la secuencia. Esta propiedad no está presente cuando no hay más nombres de sombra que devolver.
- **timestamp** fecha y hora en que AWS IoT generó la respuesta.

Propiedades del documento

El documento de sombra de un dispositivo tiene las propiedades siguientes:

state

desired

El estado deseado del dispositivo. Las aplicaciones pueden escribir en esta parte del documento para actualizar el estado de un dispositivo directamente, sin tener que conectarse al mismo.

reported

El estado notificado del dispositivo. Los dispositivos escriben en esta parte del documento para notificar su nuevo estado. Las aplicaciones leen esta parte del documento para determinar el último estado notificado del dispositivo.

metadata

Información acerca de los datos almacenados en la sección `state` del documento. Esto incluye las marcas de tiempo, según la fecha de inicio Unix, para cada atributo de la sección `state`, lo que le permite determinar cuándo se actualizaron.

Note

Los metadatos no contribuyen al tamaño de documento para establecer los límites del servicio o el precio. Para obtener más información, consulte [AWS IoT Service Limits](#).

timestamp

Indica cuándo AWS IoT envió el mensaje. Mediante el uso de la marca temporal en el mensaje y las marcas temporales para atributos individuales en la sección `desired` o `reported`, un dispositivo puede determinar la antigüedad de una propiedad, incluso si el dispositivo no tiene un reloj interno.

clientToken

Una cadena única del dispositivo que le permite asociar respuestas a solicitudes en un entorno de MQTT.

version

La versión del documento. Cada vez que el documento se actualiza, su número de versión se incrementa. Se utiliza para garantizar que la versión del documento que se actualiza sea la más reciente.

Para obtener más información, consulte [Ejemplos de documento de sombra \(p. 665\)](#).

Estado delta

El estado delta es un tipo de estado virtual que contiene la diferencia entre los estados `desired` y `reported`. Los campos de la sección `desired` que no están incluidos en la sección `reported` se incluyen en el delta. Los campos que están en la sección `reported` y no están en la sección `desired` no se incluyen en el delta. El delta contiene metadatos, y sus valores son iguales a los metadatos del campo `desired`. Por ejemplo:

```
{  
  "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  
}
```

Cuando los objetos anidados difieren, el delta contiene la ruta de acceso a la raíz.

```
{  
  "state": {  
    "desired": {  
      "lights": {  
        "color": {  
          "r": 255,  
          "g": 255,  
          "b": 255  
        }  
      }  
    }  
  }
```

```
        "g": 255,
        "b": 255
    }
},
"reported": {
    "lights": {
        "color": {
            "r": 255,
            "g": 0,
            "b": 255
        }
    }
},
"delta": {
    "lights": {
        "color": {
            "g": 255
        }
    }
},
"version": 18,
"timestamp": 123456789
}
```

El servicio Device Shadow calcula la diferencia iterando por cada campo con el estado `desired` y comparándolo con el estado `reported`.

Las matrices se tratan como valores. Si una matriz de la sección `desired` no coincide con la matriz de la sección `reported`, toda la matriz deseada se copia en el delta.

Control de versiones de documentos de sombra

El servicio Device Shadow admite el control de versiones en cada mensaje de actualización, tanto de solicitud como de respuesta. Esto significa que con cada actualización de una sombra, se incrementa la versión del documento JSON. Esto permite garantizar dos cosas:

- Un cliente puede recibir un error si intenta sobrescribir una sombra con una versión más antigua. Se informa al cliente de que debe volver a sincronizarlo para poder actualizar la sombra de un dispositivo.
- Un cliente puede decidir omitir un mensaje recibido si su versión es anterior a la que tiene almacenada.

Un cliente puede omitir la coincidencia de versiones al no incluir una versión en el documento de sombra.

Tokens de cliente en documentos de sombra

Puede utilizar un token de cliente con mensajes basados en MQTT para verificar que el mismo token de cliente esté contenido en una solicitud y en la respuesta a dicha solicitud. De esta forma se garantiza que la respuesta y la solicitud estén asociadas.

Note

El token de cliente no puede ser superior a 64 bytes. Un token de cliente que sea superior a 64 bytes provocará una respuesta 400 (solicitud errónea) y un mensaje de error `clientToken` no válido.

Propiedades de documento de sombra vacío

Las propiedades `reported` y `desired` de un documento de sombra pueden estar vacías u omitidas cuando no se aplican al estado de sombra actual. Por ejemplo, un documento de sombra contiene una

propiedad `desired` solo si tiene un estado deseado. A continuación se muestra un ejemplo válido de un documento de estado sin propiedad `desired`:

```
{  
    "reported" : { "temp": 55 }  
}
```

La propiedad `reported` también puede estar vacía, por ejemplo, si el dispositivo no ha actualizado la sombra:

```
{  
    "desired" : { "color" : "RED" }  
}
```

Si una actualización hace que las propiedades `desired` o `reported` se conviertan en nulas, se quita del documento. A continuación se muestra cómo quitar la propiedad `desired` estableciéndola en `null`. Puede hacerlo cuando un dispositivo actualice su estado, por ejemplo.

```
{  
    "state": {  
        "reported": {  
            "color": "red"  
        },  
        "desired": null  
    }  
}
```

Un documento de sombra también puede no tener ni la propiedad `desired` ni `reported`, lo que hace que el documento de sombra esté vacío. Este es un ejemplo de un documento de sombra vacío pero válido.

```
{  
}
```

Valores de matriz en documentos sombra

Las sombras son compatibles con las matrices, pero las tratan como valores normales, en el sentido de que la actualización de una matriz sustituye toda la matriz. No es posible actualizar parte de una matriz.

Estado inicial:

```
{  
    "desired" : { "colors" : [ "RED", "GREEN", "BLUE" ] }  
}
```

Actualizar:

```
{  
    "desired" : { "colors" : [ "RED" ] }  
}
```

Estado final:

```
{  
    "desired" : { "colors" : [ "RED" ] }  
}
```

Las matrices no pueden tener valores nulos. Por ejemplo, la matriz siguiente no es válida y se rechazará.

```
{  
    "desired" : {  
        "colors" : [ null, "RED", "GREEN" ]  
    }  
}
```

Mensajes de error de Device Shadow

El servicio Device Shadow publica un mensaje en el tema de errores (a través de MQTT) si se produce un error al intentar cambiar el documento de estado. Este mensaje solo se genera como respuesta a una solicitud de publicación en uno de los reservados.`$awstemas`. Si el cliente actualiza el documento mediante la API REST, recibe el código de error HTTP como parte de su respuesta y no se genera un mensaje de error MQTT.

Código de error HTTP	Mensajes de error
400 (solicitud errónea)	<ul style="list-style-type: none">JSON no válidoFalta un nodo necesario: estadoEl nodo de estado debe ser un objetoEl nodo deseado debe ser un objetoEl nodo notificado debe ser un objetoVersión no válidaClientToken no válido <p>Note</p> <p>Un token de cliente que sea mayor de 64 bytes provocará esta respuesta.</p> <ul style="list-style-type: none">JSON contiene demasiados niveles de anidamiento; el máximo permitido es 6El estado contiene un nodo no válido
401 (sin autorización)	<ul style="list-style-type: none">Sin autorización
403 (prohibido)	<ul style="list-style-type: none">Prohibido
404 (no encontrado)	<ul style="list-style-type: none">Objeto no encontradoNo existe ninguna sombra con el nombre: <code>shadowName</code>
409 (conflicto)	<ul style="list-style-type: none">Conflicto de versiones
413 (carga demasiado grande)	<ul style="list-style-type: none">La carga supera el tamaño máximo permitido
415 (tipo de medio incompatible)	<ul style="list-style-type: none">Codificación documentada incompatible, se admite la codificación UTF-8
429 (demasiadas solicitudes)	<ul style="list-style-type: none">El servicio Device Shadow generará este mensaje de error cuando haya más de diez solicitudes en vuelo en una sola conexión.
500 (error de servidor interno)	<ul style="list-style-type: none">Error de servicio interno

Trabajos

Usar AWS IoT Trabajos para definir un conjunto de operaciones remotas que se pueden enviar a uno o más dispositivos conectados a o que se ejecutan en uno o más de esos dispositivos AWS IoT. Por ejemplo, puede definir un trabajo que indique a un conjunto de dispositivos descargar e instalar aplicaciones, ejecutar actualizaciones de firmware, reiniciar, rotar certificados o realizar operaciones remotas de solución de problemas.

Para crear trabajos, defina primero un documento de trabajo que contiene una lista de instrucciones que describen las operaciones que el dispositivo debe realizar de forma remota. Para realizar estas operaciones, especifique una lista de destinos, que son cosas individuales, [grupos de objetos \(p. 273\)](#), o ambas. El documento de trabajo y los objetivos juntos constituyen un despliegue.

Cada implementación puede tener configuraciones adicionales:

- **despliegue:** Esta configuración define cuántos dispositivos reciben el documento de trabajo cada minuto.
- **Abortar:** Si un cierto número de dispositivos no recibe la notificación de trabajo, utilice esta configuración para cancelar el trabajo y evitar enviar una actualización incorrecta a toda una flota.
- **Timeout (Tiempo de espera):** Si no se recibe una respuesta de tus objetivos de trabajo dentro de un plazo determinado, el trabajo puede fallar. Puede realizar un seguimiento del trabajo que se está ejecutando en estos dispositivos.
- **Intentar de nuevo:** Si un dispositivo informa de un fallo o se agota el tiempo de espera de un trabajo, puede utilizar AWS IoT Trabajos para reenviar automáticamente el documento de trabajo al dispositivo.

Jobs de AWS IoT envía un mensaje para informar a los destinos de que el trabajo está disponible. El destino inicia la ejecución del trabajo descargando el documento de trabajo, realizando las operaciones que especifica e informando de su progreso a AWS IoT. Puede realizar un seguimiento del progreso de un trabajo para un destino específico y para todos los destinos del trabajo mediante la ejecución de comandos proporcionados por AWS IoT Trabajos. Cuando ha comenzado un trabajo, un `En curso` se informa de estado. A continuación, los dispositivos informan de actualizaciones incrementales mientras muestran este estado hasta que el trabajo se haya realizado correctamente, ha fallado o se ha agotado el tiempo de espera.

Conceptos clave de trabajos

Tarea

Un trabajo es una operación remota que se envía a uno o varios dispositivos conectados a o que se ejecuta en ellos AWS IoT. Por ejemplo, puede definir un trabajo que indique a un conjunto de dispositivos descargar e instalar una aplicación o ejecutar actualizaciones de firmware, reiniciar, rotar certificados o realizar operaciones remotas de solución de problemas.

documento de Job

Para crear un trabajo, primero debe crear un documento de trabajo, que es una descripción de las operaciones remotas que deben realizar los dispositivos.

Los documentos de trabajo son documentos JSON con codificación UTF-8 y deben contener la información que necesitan los dispositivos para realizar un trabajo. Un documento de trabajo contendrá una o más URL en las que el dispositivo puede descargar una actualización o cualquier otro

dato. El documento de trabajo puede almacenarse en un bucket de Amazon S3 o incluirse en línea con el comando que crea el trabajo.

Tip

Para ver ejemplos de documentos de trabajo, vea el ejemplo [jobs-agent.js](#) en el AWS IoT SDK para JavaScript.

Target

Cuando cree un trabajo, debe especificar una lista de destinos que son los dispositivos que deben realizar las operaciones. Los destinos pueden ser objetos, [grupos de objetos \(p. 273\)](#) o ambos. El servicio Jobs de AWS IoT envía un mensaje a cada destino para informarle de que hay un trabajo disponible.

Deployment (Implementación)

Después de crear un trabajo proporcionando el documento de trabajo y especificando la lista de destinos, el documento de trabajo se implementa en los dispositivos de destino remotos para los que desea realizar la actualización. Para los trabajos de instantáneas, el trabajo se realizará después de implementarse en los dispositivos de destino. Para trabajos continuos, un trabajo se implementa en un grupo de dispositivos a medida que se añaden a los grupos.

Trabajoejecución

Una ejecución de trabajo es una instancia de un trabajo en un dispositivo de destino. El objetivo inicia una ejecución de trabajo descargando el documento de trabajo. A continuación, realiza las operaciones especificadas en el documento e informa de su progreso a AWS IoT. Un número de ejecución es un identificador único de una ejecución de trabajo en un destino específico. El servicio Jobs proporciona comandos para realizar un seguimiento del progreso de una ejecución de trabajo en un destino y del progreso de un trabajo en todos los destinos.

trabajo de instantánea

De forma predeterminada, un trabajo se envía a todos los destinos especificados al crearlo. Después de que esos destinos finalicen el trabajo (o informen de que no pueden hacerlo), el trabajo se completa.

Continuotrabajo

Un trabajo continuo se envía a todos los destinos especificados al crearlo. Sigue ejecutándose y se envía a todos los nuevos dispositivos (objetos) que se añaden al grupo de destino. Por ejemplo, puede usarse un trabajo continuo para incorporar o actualizar dispositivos a medida que se añaden a un grupo. Puede hacer que un trabajo sea continuo estableciendo un parámetro opcional al crearlo.

Note

Al orientar su flota de IoT mediante grupos de cosas dinámicos, le recomendamos que utilice trabajos continuos en lugar de trabajos instantáneos. Al utilizar trabajos continuos, los dispositivos que se unen al grupo reciben la ejecución del trabajo incluso después de crear el trabajo.

URL prefirmadas

Para obtener acceso seguro y por tiempo limitado a los datos que no están incluidos en el documento de trabajo, puede usar URL de Amazon S3 prefirmadas. Coloque sus datos en un bucket de Amazon S3 y añada un enlace de marcador de posición para los datos del documento de trabajo. Cuando AWS IoTJobs recibe una solicitud de documento de trabajo, analiza ese documento mediante la búsqueda de los enlaces de marcador de posición y, a continuación, los sustituye por URL de Amazon S3 prefirmadas.

El enlace de marcador de posición tiene la forma siguiente:

```
${aws:iot:s3-presigned-url:https://s3.amazonaws.com/bucket/key}
```

en donde **bucket** es el nombre del bucket y **clave** es el objeto en el bucket para el que se está realizando la vinculación.

En las regiones de Beijing y Ningxia, las URL prefirmadas solo funcionan si el propietario del recurso tiene una licencia de ICP (proveedor de contenido de Internet). Para obtener más información, consulte [Amazon Simple Storage Service](#) en la Introducción a AWSServicios in China.

despliegues

Puede especificar la rapidez con la que se notifica a los destinos una implementación de trabajo pendiente. Esto le permite crear un despliegue por etapas para administrar mejor las actualizaciones, los reinicios y otras operaciones. Puede crear una configuración de despliegue utilizando una velocidad de despliegue estática o una velocidad de despliegue exponencial. Para especificar el número máximo de destinos de trabajo para informar por minuto, utilice una tasa de despliegue estática.

Para obtener ejemplos de configuración de velocidades de despliegue y para obtener más información acerca de cómo configurar despliegues de trabajos, consulte [Despliegue de trabajos y configuración de anulaciones](#).

Abortar

Puede crear un conjunto de condiciones para cancelar despliegues una vez se cumplan los criterios que especifique. Para obtener más información, consulte [Despliegue de trabajos y configuración de anulaciones](#).

Tiempos de espera

Los tiempos de espera de Job le notifican cada vez que una implementación de trabajos se atasca en `IN_PROGRESS` estado durante un período de tiempo inesperadamente largo. Existen dos tipos de temporizadores: temporizadores en curso y temporizadores de pasos. Cuando el trabajo es `IN_PROGRESS`, puede monitorizar y realizar un seguimiento del progreso de la implementación de su trabajo.

Las implementaciones y las configuraciones de cancelación son específicas de su trabajo, mientras que la configuración de tiempo de espera es específica de una implementación de trabajos. Para obtener más información, consulte [Tiempo de espera de ejecución de Job y configuraciones de reintentos \(p. 709\)](#).

Reintentos

Los reintentos de Job permiten volver a intentar la ejecución del trabajo cuando un trabajo falla o se agota el tiempo de espera o ambos. Puede tener hasta un máximo de 10 reintentos para intentar ejecutar el trabajo. Puede supervisar y realizar un seguimiento del progreso del intento de reinicio y si la ejecución del trabajo se ha realizado correctamente.

Las configuraciones de implementación y cancelación son específicas de su trabajo, mientras que las configuraciones de tiempo de espera y reinicio son específicas de la ejecución de un trabajo. Para obtener más información, consulte [Tiempo de espera de ejecución de Job y configuraciones de reintentos \(p. 709\)](#).

Administrar trabajos

Utilice trabajos para notificar a los dispositivos de una actualización de software o de firmware. Puede utilizar la [AWS IoT consola](#), la [Operaciones de API de administración y control de Job \(p. 734\)](#), la [AWS Command Line Interface](#), o la [AWSSDK](#) para crear y administrar trabajos.

Firma de código para trabajos

Al enviar código a los dispositivos, para que los dispositivos detecten si el código se ha modificado en tránsito, le recomendamos que firme el archivo de código mediante la AWS CLI. Para obtener instrucciones, consulte [??? \(p. 680\)](#).

Para obtener más información, consulte [Para qué sirve la firma de códigoAWS IoT?](#).

documento de Job

Antes de crear un trabajo debe crear un documento del trabajo. Si está utilizando la firma de código para AWS IoT, debe cargar su documento de trabajo en un bucket de Amazon S3 con control de versiones. Para obtener más información acerca de la creación de un bucket de Amazon S3 y cargar archivos en él, consulte [Introducción a Amazon Simple Storage Service](#) en la Guía de introducción a Amazon S3.

Tip

Para ver ejemplos de documentos de trabajo, vea el ejemplo [jobs-agent.js](#) en el AWS IoT SDK para JavaScript.

URL prefirmadas

El documento de trabajo puede contener una URL de Amazon S3 prefirmada que apunte a su archivo de código (u otro archivo). Las URL de Amazon S3 prefirmadas son válidas solo durante un tiempo limitado y se generan cuando un dispositivo solicita un documento de trabajo. Dado que la URL prefirmada no se crea cuando crea el documento de trabajo, utilice un URL de marcador de posición en su documento de trabajo. Una URL de marcador de posición tiene el siguiente aspecto:

```
 ${aws:iot:s3-presigned-url:https://s3.region.amazonaws.com/<bucket>/<code file>}
```

donde:

- *balance* es el bucket de Amazon S3 que contiene el archivo de código.
- *archivo de código* es la clave de Amazon S3 del archivo de código.

Cuando un dispositivo solicita el documento de trabajo, AWS IoT genera la URL prefirmada y sustituye la URL de marcador de posición por la URL prefirmada. El documento del trabajo se envía al dispositivo.

Función de IAM para conceder permiso para descargar archivos desde S3

Al crear un trabajo que usa URL de Amazon S3 prefirmadas, debe proporcionar un rol de IAM que conceda permiso para descargar archivos desde el bucket de Amazon S3 donde se guardan los datos o las actualizaciones. El rol debe conceder permiso también para que AWS IoT asuma el rol.

Puede especificar un tiempo de espera opcional para la URL prefirmada. Para obtener más información, consulte [CreateJob](#).

Conceder AWS IoT Permiso de trabajo para asumir su rol

1. Vaya a [Centro de roles de la consola de IAM](#) y elija su rol.
2. En la página Relaciones de confianza pestaña, elija Editar relación de confianza y sustituya el documento de política por el JSON siguiente. Elija Update Trust Policy (Actualizar política de confianza).

```
{
```

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

- Si su trabajo utiliza un documento de trabajo que es un objeto Amazon S3, elijaPermisos con el JSON siguiente, añada una política que conceda permiso para descargar archivos desde su bucket de Amazon S3:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "s3:GetObject",
            "Resource": "arn:aws:s3:::your_S3_bucket/*"
        }
    ]
}
```

Temas

- [Cree y administre trabajos de mediante elAWS Management Console \(p. 679\)](#)
- [Cree y administre trabajos de mediante elAWS CLI \(p. 680\)](#)

Cree y administre trabajos de mediante elAWS Management Console

Para crear un trabajo de

- Elija su tipo de trabajo

- Vaya a [Centro de Job delAWS IoTconsola](#) y eligeCrear el trabajo.
- Según el dispositivo que esté utilizando, puede crear un trabajo personalizado, un trabajo de actualización de FreeRTOS OTA o unAWS IoT Greengrasstrabajo. En este ejemplo, elijaCreación de un trabajo personalizado.

- Introducir las propiedades de trabajo

Introduzca un nombre de trabajo alfanumérico único, una descripción y etiquetas opcionales y, a continuación, elijaPróximo.

Note

Le recomendamos que no utilice información personal en las descripciones y ID de trabajo.

- Elija sus objetivos

Elija como objetivo el trabajo a las cosas o grupos de cosas que desea ejecutar en este trabajo.

4. Especifique el documento de trabajo

Puede cargar el archivo de trabajo JSON en un depósito de S3 y, a continuación, utilizarlo como documento de trabajo o elegir el archivo de trabajo de una plantilla.

Si utilizas una plantilla, puedes elegir entre una plantilla de trabajo personalizada o una AWSplantilla administrada. Si vas a crear un trabajo para realizar acciones remotas de uso frecuente, como reiniciar el dispositivo, puedes usar una AWSplantilla administrada. Estas plantillas ya han sido preconfiguradas para su uso. Para obtener más información, consulte [Creación de una plantilla de trabajo personalizada \(p. 702\)](#) y [Crear plantillas de trabajo personalizadas a partir de plantillas administradas \(p. 698\)](#).

5. Elija su tipo de trabajo

En la páginaConfiguración de Job, elija el tipo de trabajo comocontinuo o un trabajo de instantánea. Un trabajo de instantáneas se completa cuando finaliza su ejecución en los dispositivos y grupos de destino. Un trabajo continuo se aplica a los grupos de cosas y se ejecuta en cualquier dispositivo que añada posteriormente a un grupo de destino especificado.

6. Especificar configuraciones adicionales (opcional)

Siga añadiendo configuraciones adicionales para su trabajo y, a continuación, revise y cree su trabajo. Para obtener información acerca de las configuraciones adicionales, consulte:

- [Despliegue de trabajos y configuraciones de anulaciones \(p. 708\)](#)
- [Tiempo de espera de ejecución de Job y configuraciones de reintentos \(p. 709\)](#)

Después de crear el trabajo, la consola genera una firma JSON y la coloca en su documento del trabajo. Puede usar la [consola de AWS IoT](#) para ver el estado de un trabajo, cancelarlo o eliminarlo. Para administrar los trabajos, vaya a la[Centro de Job de la consola](#).

Cree y administre trabajos de mediante elAWS CLI

En esta sección se describe cómo crear y administrar trabajos.

Creación de trabajos

Para crear unAWS IoT trabajo, utilice elCreateJobcomando. El trabajo se pone en cola para la ejecución en los destinos (objetos o grupos de objetos) que especifique. Para crear unAWS IoT job, necesita un documento de trabajo que pueda incluirse en el cuerpo de la solicitud o como un enlace a un documento de Amazon S3. Si el trabajo incluye la descarga de archivos mediante URL de Amazon S3 prefijadas, necesita un rol de IAM de Amazon Resource Name (ARN) que tenga permiso para descargar el archivo y conceda permiso a laAWS IoT Servicio de trabajos para asumir el rol.

Firma de código con trabajos

Si utilizas la firma de código paraAWS IoT, debe iniciar un trabajo de firma con código e incluir la salida en su documento de trabajo. Usar[start-signing-job](#)para crear un trabajo de firma de código.[start-signing-job](#)devuelve un ID de trabajo. Para obtener la ubicación de Amazon S3 donde se almacena la firma, utilice [ladescribe-signing-job](#)comando. Después podrá descargar la firma de Amazon S3. Para obtener más información acerca de las tareas de firma con código, consulte[Firma de código paraAWS IoT](#).

El documento del trabajo debe contener un marcador de posición de URL prefijada para su archivo de código y el resultado de la firma JSON en un bucket de Amazon S3 utilizando la[start-signing-job](#)comando, incluido en uncodesignelemento:

```
{  
  "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>  
    }  
}  
}
```

Creación de un trabajo con un documento de trabajo

El comando siguiente muestra cómo crear un trabajo utilizando un documento de trabajo ([job-document.json](#)) almacenado en un bucket de Amazon S3 ([jobBucket](#)), y un rol con permiso para descargar archivos de Amazon S3 ([Función de descarga de S3](#)).

```
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}"
```

El trabajo se ejecuta en[cosa thingOne](#).

El parámetro `timeout-config` opcional especifica la cantidad de tiempo que cada dispositivo tiene para finalizar su ejecución del trabajo. El temporizador comienza cuando el estado de ejecución del trabajo se establece en `IN_PROGRESS`. Si el estado de ejecución del trabajo no se establece en otro estado terminal antes de que pase el plazo, se establecerá en`TIMED_OUT`.

El temporizador en curso no se puede actualizar y se aplica a todas las ejecuciones de trabajos para el trabajo. Siempre que la ejecución de un trabajo permanezca en el `IN_PROGRESS` estado durante un periodo superior a este intervalo, falla y cambia al terminal `TIMED_OUT` estado. AWS IoT también publica una notificación MQTT.

Para obtener más información acerca de la creación de configuraciones para implementaciones y anulaciones de trabajos, consulte[Despliegue de trabajos y configuración de anulaciones](#).

Note

Los documentos de Job que se especifican como archivos de Amazon S3 se recuperan en el momento en el que crea el trabajo. Si cambia el contenido del archivo de Amazon S3 que usó como origen de su documento de trabajo después de haber creado el documento de trabajo, entonces lo que se envía a los destinos del trabajo no cambiará.

Actualización de un trabajo

Para actualizar un trabajo, utilice el `UpdateJob` comando. Puede actualizar los campos `description`, `presignedUrlConfig`, `jobExecutionsRolloutConfig`, `abortConfig` y `timeoutConfig` para un trabajo.

```
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 obtener más información, consulte [Despliegue de trabajos y configuración de anulaciones](#).

Cancelación de un trabajo

Para cancelar un trabajo, utilice el `CancelJob` comando. Al cancelar un trabajo, AWS IoT dejará de desplegar nuevas ejecuciones de trabajos para el trabajo. También cancela cualquier ejecución de trabajo que se encuentre en `UNQUEUED` estado. AWS IoT mantiene las ejecuciones de trabajo con un estado terminal intactas porque el dispositivo ya ha completado el trabajo. Si el estado de la ejecución de un trabajo es `IN_PROGRESS`, también se mantendrá intacta a menos que se use el parámetro opcional `--force`.

El siguiente comando muestra cómo cancelar un trabajo con ID 010.

```
aws iot cancel-job --job-id 010
```

El comando muestra el resultado siguiente:

```
{  
  "jobArn": "string",  
  "jobId": "string",  
  "description": "string"  
}
```

Cuando se cancela un trabajo, se cancelan las ejecuciones de trabajos con estado `QUEUED`. Ejecuciones de Job que están en `IN_PROGRESS` estado se cancelarán, pero solo si especifica el opcional `--force` parámetro. Las ejecuciones de Job con un estado terminal no se cancelarán.

Warning

Cancelación de un trabajo que se encuentra en el `IN_PROGRESS` state (configurando el `--force`) cancela las ejecuciones de trabajos en curso y hará que el dispositivo que está ejecutando el trabajo no pueda actualizar el estado de ejecución del trabajo. Sea precavido y asegúrese de que cada dispositivo que ejecute un trabajo cancelado pueda recuperarse a un estado válido.

El estado de un trabajo cancelado o de una de sus ejecuciones de trabajos es a la larga coherente: AWS IoT detiene lo antes posible la programación de nuevas ejecuciones de trabajos y las ejecuciones en dispositivos de trabajos con estado `QUEUED`. Cambiar el estado de la ejecución de un trabajo a `CANCELED` puede llevar algo de tiempo, según el número de dispositivos y otros factores.

Si un trabajo se cancela porque cumple los criterios definidos por un objeto `AbortConfig`, el servicio añade valores rellenados automáticamente para los campos `reasonCode` y `comment`. Puede crear sus propios valores para `reasonCode` cuando el trabajo se cancela por iniciativa del usuario.

Cancelación de una ejecución de trabajo

Para cancelar la ejecución de un trabajo en un dispositivo, utilice la `CancelJobExecution` comando. Cancela una ejecución de trabajo que está en `UNQUEUED` estado. Si desea cancelar la ejecución de un trabajo en curso, debe utilizar la `--force` parámetro.

El siguiente comando muestra cómo cancelar la ejecución de un trabajo del trabajo 010 que se ejecuta en `myThing`.

```
aws iot cancel-job-execution --job-id 010 --thing-name myThing
```

El comando no muestra ninguna salida.

Una ejecución de trabajo que está en `UNQUEUED` El estado se ha cancelado. Una ejecución de trabajo que está en `UNIN_PROGRESS` estado se cancela, pero solo si especifica la opción `--force` parámetro. Las ejecuciones de Job con un estado terminal no se pueden cancelar.

Warning

Cuando se cancela la ejecución de un trabajo en `IN_PROGRESS` estado, el dispositivo no puede actualizar el estado de ejecución del trabajo. Sea precavido y Asegúrese de que el dispositivo puede recuperarse a un estado válido.

Si la ejecución del trabajo se encuentra en estado terminal o si la ejecución del trabajo se encuentra en `UNIN_PROGRESS` state y el `--force` El parámetro no se establece `true`, este comando provoca un `InvalidStateTransitionException`.

El estado de la ejecución de un trabajo cancelada es a la larga coherente. Cambio del estado de una ejecución de trabajo `ACANCELED` puede llevar algo de tiempo, según varios factores.

Eliminación de un trabajo

Para eliminar un trabajo y sus ejecuciones de trabajo, utilice la `DeleteJob` comando. De forma predeterminada, solo puede eliminar un trabajo con estado terminal (`SUCCEEDED` o `CANCELED`). En caso contrario, se produce una excepción. Puede eliminar un trabajo en `IN_PROGRESS` indique, sin embargo, si el `force` El parámetro de se establece `true`.

Para eliminar un trabajo, ejecute el siguiente comando:

```
aws iot delete-job --job-id 010 --force|--no-force
```

El comando no muestra ninguna salida.

Warning

Cuando eliminas un trabajo que se encuentra en `IN_PROGRESS`, el dispositivo que está implementando el trabajo no puede obtener acceso a la información del trabajo ni actualizar el estado de ejecución del trabajo. Actúe con precaución y asegúrese de que cada dispositivo que despliegue un trabajo eliminado pueda recuperarse a un estado válido.

Eliminar un trabajo podría llevar algún tiempo, en función del número de ejecuciones de trabajos creadas para el trabajo y otros factores. Aunque el trabajo se esté eliminando, como estado del trabajo se muestra `DELETION_IN_PROGRESS`. Si se intenta eliminar o cancelar un trabajo con un estado que ya está, se producirá un error `DELETION_IN_PROGRESS`.

Solo puede haber 10 trabajos con estado `DELETION_IN_PROGRESS` al mismo tiempo. De lo contrario, se genera `LimitExceeded` exception.

Obtención de un documento de trabajo

Para recuperar un documento de trabajo para un trabajo, utilice la `GetJobDocument` comando. Un documento de trabajo es una descripción de las operaciones remotas que deben ejecutar los dispositivos.

Para obtener un documento de trabajo, ejecute el siguiente comando:

```
aws iot get-job-document --job-id 010
```

El comando devuelve el documento de trabajo para el trabajo especificado:

```
{  
    "document": "{\n\t\"operation\":\"install\",\\n\t\"url\":\"http://amazon.com/  
firmWareUpdate-01\",\\n\t\"data\":\"\${aws:iot:s3-presigned-url:https://s3.amazonaws.com/job-  
test-bucket/datafile}\\n\"}  
}
```

Note

Cuando utiliza este comando para recuperar un documento de trabajo, las URL de marcador de posición no se sustituirán por URL de Amazon S3 prefirmadas. Cuando un dispositivo llama al `GetPendingJobExecutions` Operación de API, las URL de marcador de posición se sustituyen por URL de Amazon S3 prefirmadas en el documento de trabajo.

Enumeración de trabajos

Para obtener una lista de todos los trabajos de Cuenta de AWS, utilice el `ListJobs` comando. Los datos del trabajo y los datos de ejecución del trabajo se conservan durante un [tiempo limitado](#). Ejecute el siguiente comando para listar todos los trabajos de su Cuenta de AWS:

```
aws iot list-jobs
```

El comando devuelve todos los trabajos en su cuenta ordenados por estado del trabajo:

```
{  
    "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,  
            "completedAt": 1486678850.575  
        }  
    ]  
}
```

```
        "createdAt": 1486678850.575,
        "targetSelection": "SNAPSHOT",
        "jobId": "011"
    ]
}
```

Descripción de un trabajo

Para obtener el estado de un trabajo, ejecute el `DescribeJob` comando. El siguiente comando muestra cómo describir un trabajo:

```
$ aws iot describe-job --job-id 010
```

El comando devuelve el estado de un trabajo especificado. Por ejemplo:

```
{
    "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
}
}
```

Enumeración de ejecuciones para un trabajo

Un trabajo que se ejecuta en un dispositivo específico se representa mediante un objeto de ejecución de trabajo. Ejecute el comando `ListJobExecutionsForJob` para enumerar todas las ejecuciones de trabajo para un trabajo. A continuación se muestra cómo enumerar las ejecuciones para un trabajo:

```
aws iot list-job-executions-for-job --job-id 010
```

El comando devuelve una lista de ejecuciones de trabajo:

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

Enumeración de ejecuciones de trabajo para un objeto

Ejecute el comando `ListJobExecutionsForThing` para enumerar todas las ejecuciones de trabajo que se ejecutan en un objeto. A continuación se muestra cómo listar las ejecuciones de trabajos para un objeto:

```
aws iot list-job-executions-for-thing --thing-name thingOne
```

El comando devuelve una lista de ejecuciones de trabajo que se ejecutan o se han ejecutado en el objeto especificado:

```
{
```

```
"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"
}
]
}
```

Descripción de una ejecución de trabajo

Ejecute el comando `DescribeJobExecution` para obtener el estado de la ejecución de un trabajo. Debe especificar un ID de trabajo y el nombre del objeto y, opcionalmente, un número de ejecución para identificar la ejecución del trabajo. A continuación se muestra cómo describir la ejecución de un trabajo:

```
aws iot describe-job-execution --job-id 017 --thing-name thingOne
```

El comando devuelve `JobExecution`. Por ejemplo:

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

Eliminación de una ejecución de trabajo

Ejecute el comando `DeleteJobExecution` para eliminar la ejecución de un trabajo. Debe especificar un ID de trabajo, un nombre de objeto y un número de ejecución para identificar la ejecución del trabajo. A continuación se muestra cómo eliminar la ejecución de un trabajo:

```
aws iot delete-job-execution --job-id 017 --thing-name thingOne --execution-number
1234567890 --force|--no-force
```

El comando no muestra ninguna salida.

De forma predeterminada, el estado de ejecución de los trabajos debe ser `QUEUED` o un estado final (`SUCCEEDED`, `FAILED`, `REJECTED`, `TIMED_OUT`, `REMOVED` o `CANCELED`). En caso contrario, se produce un error. Para eliminar la ejecución de un trabajo con estado `IN_PROGRESS`, puede establecer el parámetro `force` en `true`.

Warning

Cuando se elimina la ejecución de un trabajo con estado `IN_PROGRESS`, el dispositivo que está ejecutando el trabajo no puede obtener acceso a la información del trabajo ni actualizar el estado de su ejecución. Actúe con precaución y asegúrese de que el dispositivo pueda recuperarse a un estado válido.

Plantillas de trabajo

Utilice plantillas de trabajos para preconfigurar los trabajos que puede implementar en varios conjuntos de dispositivos de destino. Para las acciones remotas que se realizan con frecuencia que desea implementar en sus dispositivos, como reiniciar o instalar una aplicación, puede utilizar plantillas para definir configuraciones estándar. Cuando desee realizar operaciones como la implementación de parches de seguridad y correcciones de errores, también puede crear plantillas a partir de trabajos existentes.

Al crear una plantilla de trabajo, puede especificar las siguientes configuraciones y recursos adicionales.

- Propiedades de Job
- Documentos de Job y objetivos
- Criterios de despliegue y cancelación
- Criterios de inactividad

Personalizado y AWSplantillas administradas

Según la acción remota que deseé realizar, puede crear una plantilla de trabajo personalizada o utilizar unAWSplantilla administrada. Utilice plantillas de trabajo personalizadas para proporcionar su

propio documento de trabajo personalizado y crear trabajos reutilizables para desplegarlos en sus dispositivos. AWS las plantillas gestionadas son plantillas de trabajo proporcionadas por AWS IoT. Los trabajos para acciones que se realizan habitualmente. Estas plantillas tienen un documento de trabajo predefinido para algunas acciones remotas, por lo que no tiene que crear su propio documento de trabajo. Las plantillas administradas le ayudan a crear trabajos reutilizables para una ejecución más rápida en sus dispositivos.

Temas

- [UsarAWSplantillas gestionadas para implementar operaciones remotas comunes \(p. 689\)](#)
- [Creación de plantillas de trabajo personalizadas \(p. 702\)](#)

UsarAWSplantillas gestionadas para implementar operaciones remotas comunes

AWS las plantillas gestionadas son plantillas de trabajo proporcionadas por AWS para acciones remotas que se realizan con frecuencia, como reiniciar, descargar un archivo o instalar una aplicación en los dispositivos. Estas plantillas tienen un documento de trabajo predefinido para cada acción remota, por lo que no tiene que crear su propio documento de trabajo.

Puede elegir entre un conjunto de configuraciones predefinidas y crear trabajos utilizando estas plantillas sin escribir ningún código adicional. Mediante plantillas administradas, puede ver el documento de trabajo implementado en sus flotas. Puede crear un trabajo utilizando estas plantillas y crear una plantilla de trabajo personalizada que pueda reutilizar para sus operaciones remotas.

¿Qué contienen las plantillas administradas de?

Cada AWS plantilla administrada contiene:

- Documento de trabajo que especifica el nombre de la operación y sus parámetros. Por ejemplo, si usa un Descargar archivoplantilla, el nombre de la operación es Descargar archivo y los parámetros pueden ser:
 - La URL del archivo que desea descargar en su dispositivo, que puede ser un recurso de Internet o una URL pública o de S3 prefirmada.
 - Ruta de archivo local en el dispositivo para almacenar el archivo descargado.

Para obtener más información acerca de los documentos de la tarea y sus parámetros, consulte [Acciones remotas de plantillas gestionadas y documentos de trabajo \(p. 690\)](#).

- Entorno para ejecutar los comandos del documento de trabajo.

Requisitos previos

Para que los dispositivos ejecuten las acciones remotas especificadas por el documento de trabajo de plantilla administrada, debe:

- Instala el software específico en tu dispositivo
- Usar AWS IoT Cliente de dispositivo

Recomendamos que la instale y la ejecute la AWS IoT Device Client en sus dispositivos porque admite el uso de todas las plantillas administradas directamente desde la consola de forma predeterminada.

Device Client es un software de código abierto escrito en C++ que puede compilar e instalar en sus dispositivos IoT basados en Linux integrados. El cliente de dispositivo tiene un cliente

base de funciones de cliente. El cliente base establece la conectividad con AWS IoT a través del protocolo MQTT y puede conectarse con las diferentes funciones del lado del cliente.

Para realizar operaciones remotas en los dispositivos, utilice el [Función Jobs de lado del cliente](#) del Device Client. Esta función contiene un analizador para recibir el documento de trabajo y los gestores de trabajos que implementan las acciones remotas especificadas en el documento de trabajo.

Para obtener más información acerca del cliente de dispositivo y sus características, consulte [AWS IoT Cliente del dispositivo](#).

Cuando se ejecuta en dispositivos, Device Client recibe el documento de trabajo y tiene una implementación específica de la plataforma que utiliza para ejecutar comandos en el documento. Para obtener más información acerca de la configuración del cliente de dispositivo y el uso de la característica Trabajos, consulte [Tutorial de AWS IoT \(p. 124\)](#).

- Utilice su propio software de dispositivo y gestores de trabajos

Alternativamente, puede escribir su propio código para los dispositivos mediante la [AWS IoT SDK de dispositivos](#) y su biblioteca de controladores que admite las operaciones remotas. Para implementar y ejecutar trabajos, asegúrese de que las bibliotecas de agentes de dispositivos se han instalado correctamente y se están ejecutando en estos dispositivos.

También puede optar por usar sus propios controladores que pueden admitir las operaciones remotas.

Para obtener más información acerca de cómo crear estos controladores, consulte [Controladores de trabajos de ejemplo en el AWS IoT Repository de GitHub de cliente de dispositivo](#).

- Utilización de un entorno compatible

Para cada plantilla administrada, encontrará información sobre el entorno que puede utilizar para ejecutar las acciones remotas. Le recomendamos que use la plantilla con un entorno Linux compatible tal y como se especifica en la plantilla. Puede utilizar el [AWS IoT Device Client](#) para ejecutar las acciones remotas de la plantilla de administración porque admite microprocesadores comunes y entornos Linux, como Debian y Ubuntu.

Acciones remotas de plantillas gestionadas y documentos de trabajo

En la siguiente sección se enumeran las diferentes [AWS plantillas administradas](#) para AWS IoT. Ejecuta y describe las acciones remotas que se pueden realizar en los dispositivos. En la sección siguiente, encontrará información sobre el documento de trabajo y una descripción de los parámetros del documento de trabajo para cada acción remota. El software del lado del dispositivo utiliza el nombre de la plantilla y los parámetros para realizar la acción remota.

AWS las plantillas gestionadas aceptan parámetros de entrada para los que se especifica un valor al crear un trabajo mediante la plantilla. Todas las plantillas administradas tienen dos parámetros de entrada opcionales en común: `runAsUser` y `pathToHandler`. Excepto por el `AWS-Reboot`, las plantillas requieren parámetros de entrada adicionales para los que debe especificar un valor al crear un trabajo mediante la plantilla. Estos parámetros de entrada necesarios varían en función de la plantilla que elija. Por ejemplo, si elige la `AWS-Download-File`, debe especificar una lista de paquetes para instalar y una URL desde la que descargar archivos.

Especifica un valor para los parámetros de entrada cuando se usa la AWS IoT consola de AWS CLI para crear un trabajo que utilice una plantilla administrada. Cuando se usa la CLI, proporciona estos valores mediante el `document-parameters` objeto. Para obtener más información, consulte [documentParameters](#).

Note

Debe utilizar `document-parameters` solo cuando se crean trabajos desde [AWS plantillas administradas](#). Este parámetro no se puede utilizar con plantillas de trabajo personalizadas ni para crear trabajos a partir de ellas.

A continuación se muestra una descripción de los parámetros de entrada opcionales comunes. Verás una descripción de otros parámetros de entrada que requiere cada plantilla administrada en la sección siguiente.

runAsUser

Este parámetro especifica si se debe ejecutar el gestor de trabajos como otro usuario. Si no se especifica durante la creación del trabajo, el gestor de trabajos se ejecuta como el mismo usuario que el cliente de dispositivo. Cuando ejecute el gestor de trabajos como otro usuario, especifique un valor de cadena que no supere los 256 caracteres.

pathToHandler

La ruta del controlador de trabajo que se ejecuta en el dispositivo. Si no se especifica durante la creación de trabajos, Device Client utiliza el directorio de trabajo actual.

A continuación se muestran las distintas acciones remotas, sus documentos de trabajo y los parámetros que aceptan. Todas estas plantillas admiten el entorno Linux para ejecutar la operación remota en el dispositivo.

[AWS: Reiniciar](#)

Nombre de la plantilla

AWS-Reboot

Descripción de la plantilla

Plantilla administrada proporcionada por AWS para reiniciar el dispositivo.

Parámetros de entrada

Esta plantilla no tiene parámetros obligatorios. Puede especificar los parámetros opcionales `runAsUser` y `pathToHandler`.

Comportamiento de

El dispositivo se reinicia correctamente.

documento de Job

A continuación se muestra el documento de trabajo y su última versión. La plantilla muestra la ruta del controlador de trabajos y el script de shell, `reboot.sh`, que el controlador de trabajo debe ejecutar para reiniciar el 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—Descargar—Archivo

Nombre de la plantilla

AWS-Download-File

Descripción de la plantilla

Plantilla administrada proporcionada por AWS para descargar un archivo.

Parámetros de entrada

Esta plantilla tiene los siguientes parámetros obligatorios. También puede especificar los parámetros opcionales `runAsUser` y `pathToHandler`.

`downloadUrl`

URL desde la que descargar el archivo, que puede ser un recurso de Internet, un objeto de Amazon S3 al que se puede acceder públicamente o un objeto de Amazon S3 al que solo puede acceder su dispositivo mediante una URL prefijada. Para obtener más información acerca de cómo usar URL prefijadas y conceder permisos, consulte [URL prefijadas \(p. 678\)](#).

`filePath`

Ruta de archivo local que muestra la ubicación en el dispositivo para almacenar el archivo descargado.

Comportamiento de

El dispositivo descarga el archivo desde la ubicación especificada, verifica que la descarga se ha completado y lo almacena localmente.

documento de Job

A continuación se muestra el documento de trabajo y su última versión. La plantilla muestra la ruta del controlador de trabajos y el script de shell `download-file.sh`, que el gestor de tareas debe ejecutar para descargar el archivo. También muestra los parámetros obligatorios `downloadUrl` y `filePath`.

```
{  
  "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—Aplicación

Nombre de la plantilla

AWS-Install-Application

Descripción de la plantilla

Plantilla administrada proporcionada por AWS para instalar una o más aplicaciones.

Parámetros de entrada

Esta plantilla tiene el siguiente parámetro obligatorio,**packages**. También puede especificar los parámetros opcionales **runAsUser** y **pathToHandler**.

packages

Una lista separada por espacios de una o varias aplicaciones que se van a instalar.

Comportamiento de

El dispositivo instala las aplicaciones tal y como se especifica en el documento de trabajo.

documento de Job

A continuación se muestra el documento de trabajo y su última versión. La plantilla muestra la ruta del controlador de trabajos y el script de shell,**install-packages.sh**, que el gestor de tareas debe ejecutar para descargar el archivo. También muestra el parámetro requerido,**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—Remove—Aplicación

Nombre de la plantilla

AWS-Remove-Application

Descripción de la plantilla

Plantilla administrada proporcionada porAWSpara desinstalar una o más aplicaciones.

Parámetros de entrada

Esta plantilla tiene el siguiente parámetro obligatorio,`packages`. También puede especificar los parámetros opcionales`runAsUser`y`pathToHandler`.

`packages`

Una lista separada por espacios de una o varias aplicaciones que se van a desinstalar.

Comportamiento de

El dispositivo desinstala las aplicaciones tal y como se especifica en el documento de trabajo.

documento de Job

A continuación se muestra el documento de trabajo y su última versión. La plantilla muestra la ruta del controlador de trabajos y el script de shell,`remove-packages.sh`, que el gestor de tareas debe ejecutar para descargar el archivo. También muestra el parámetro requerido`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—Aplicación

Nombre de la plantilla

`AWS-Start-Application`

Descripción de la plantilla

Plantilla administrada proporcionada porAWSPara iniciar uno o varios servicios.

Parámetros de entrada

Esta plantilla tiene el siguiente parámetro obligatorio,`services`. También puede especificar los parámetros opcionales`runAsUser`y`pathToHandler`.

`services`

Una lista separada por espacios de una o varias aplicaciones que se van a iniciar.

Comportamiento de

Las aplicaciones especificadas comienzan a ejecutarse en el dispositivo.

documento de Job

A continuación se muestra el documento de trabajo y su última versión. La plantilla muestra la ruta del controlador de trabajos y el script de shell,`start-services.sh`, que el gestor de tareas debe ejecutar para iniciar los servicios del sistema. También muestra el parámetro requerido.`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 — Aplicación

Nombre de la plantilla

`AWS-Stop-Application`

Descripción de la plantilla

Plantilla administrada proporcionada por AWSPara detener uno o varios servicios.

Parámetros de entrada

Esta plantilla tiene el siguiente parámetro obligatorio,`services`. También puede especificar los parámetros opcionales`runAsUser`y`pathToHandler`.

`services`

Una lista separada por espacios de una o varias aplicaciones que se van a detener.

Comportamiento de

Las aplicaciones especificadas dejan de ejecutarse en el dispositivo.

documento de Job

A continuación se muestra el documento de trabajo y su última versión. La plantilla muestra la ruta del controlador de trabajos y el script de shell,`stop-services.sh`, que el gestor de tareas debe ejecutar para detener los servicios del sistema. También muestra el parámetro requerido.`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 — Aplicación

Nombre de la plantilla

AWS-Restart-Application

Descripción de la plantilla

Plantilla administrada proporcionada por AWS para detener y reiniciar uno o varios servicios.

Parámetros de entrada

Esta plantilla tiene el siguiente parámetro obligatorio,**services**. También puede especificar los parámetros opcionales**runAsUser** y **pathToHandler**.

services

Una lista separada por espacios de una o varias aplicaciones que se van a reiniciar.

Comportamiento de

Las aplicaciones especificadas se detienen y luego se reinician en el dispositivo.

documento de Job

A continuación se muestra el documento de trabajo y su última versión. La plantilla muestra la ruta del controlador de trabajos y el script de shell,**restart-services.sh**, que el gestor de tareas debe ejecutar para reiniciar los servicios del sistema. También muestra el parámetro requerido.**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}"  
    }  
}  
]
```

Temas

- [Creación de un trabajo desdeAWSplantillas gestionadas mediante elAWS Management Console \(p. 697\)](#)
- [Crear un trabajo deAWSplantillas gestionadas mediante elAWS CLI \(p. 699\)](#)

Creación de un trabajo desdeAWSplantillas gestionadas mediante elAWS Management Console

UsarAWS Management ConsolePara obtener información sobreAWSplantillas gestionadas y cree un trabajo utilizando estas plantillas. A continuación, puede guardar el trabajo que cree como su propia plantilla personalizada.

Obtener información detallada sobre plantillas administradas

Puede obtener información sobre las diferentes plantillas administradas que están disponibles para usar en elAWS IoTconsola de .

1. Para ver las plantillas administradas disponibles, vaya a la[Centro de plantillas de Job delAWS IoTconsola](#)y elige elPlantillas administradasPestaña.
2. Elija una plantilla administrada para ver sus detalles.

La página de detalles contiene la siguiente información:

- Nombre, descripción y nombre de recurso de Amazon (ARN) de la plantilla administrada.
- Entorno en el que se pueden realizar las operaciones remotas, como Linux.
- El documento de trabajo JSON que especifica la ruta al gestor de trabajos y los comandos que se ejecutarán en el dispositivo. Por ejemplo, a continuación se muestra un documento de trabajo de ejemplo para elReinicio de AWSplantilla de. La plantilla muestra la ruta del controlador de trabajos y el script de shell,reboot.sh, que el controlador de trabajo debe ejecutar para reiniciar el 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 obtener más información acerca del documento de trabajo y sus parámetros para varias acciones remotas, consulte[Acciones remotas de plantillas gestionadas y documentos de trabajo \(p. 690\)](#).

- La última versión del documento de trabajo.

Crear un trabajo mediante plantillas administradas

Puede utilizar la consola de para elegir unAWSplantilla administrada para usar para crear un trabajo. En esta sección le demostramos cómo.

También puede iniciar el flujo de trabajo de creación de trabajos y, a continuación, elegir laAWSplantilla administrada que desea utilizar durante la creación del trabajo. Para obtener más información acerca de este flujo de trabajo, consulte[Cree y administre trabajos de mediante elAWS Management Console \(p. 679\)](#).

1. Elija suAWSplantilla administrada

Vaya a[Centro de plantillas de Job delAWS IoTconsola](#), elige elPlantillas administradasy, a continuación, elija su plantilla.

2. Crear un trabajo utilizando la plantilla administrada

1. En la página de detalles de la plantilla, eligeCrear el trabajo.

La consola cambia alPropiedades de trabajo personalizadaspaso delCrear el trabajo flujo de trabajo en el que se ha añadido la configuración de la plantilla.

2. Introduzca un nombre de trabajo alfanumérico único y una descripción y etiquetas opcionales y, a continuación, elijaPróximo.

3. Elija las cosas o grupos de cosas como objetivos de trabajo que desea ejecutar en este trabajo.

4. En el navegador documento de Job, la plantilla se muestra con sus ajustes de configuración y parámetros de entrada. Introduzca valores para los parámetros de entrada de la plantilla elegida. Por ejemplo, si eligió laAWS-Descargar archivoplantilla:

- ParaURL de descarga, escriba la URL del archivo que desea descargar, por ejemplo:<https://example.com/index.html>.

- ParafilePath, introduzca la ruta del dispositivo para almacenar el archivo descargado, por ejemplo:[path/to/file](#).

Si lo desea, también puede escribir valores para los parámetros[runAsUser](#) y [pathToHandler](#).

Para obtener más información acerca de los parámetros de entrada de cada plantilla, consulte[Acciones remotas de plantillas gestionadas y documentos de trabajo \(p. 690\)](#).

5. En la páginaConfiguración de Job, elija el tipo de trabajo comocontinuo o un trabajo de instantánea. Un trabajo de instantáneas se completa cuando finaliza su ejecución en los dispositivos y grupos de destino. Un trabajo continuo se aplica a los grupos de cosas y se ejecuta en cualquier dispositivo que agregue a un grupo de destino especificado.

6. Siga añadiendo configuraciones adicionales para su trabajo y, a continuación, revise y cree su trabajo. Para obtener más información acerca de las configuraciones adicionales, consulte:

- [Despliegue de trabajos y configuraciones de anulaciones \(p. 708\)](#)

- [Tiempo de espera de ejecución de Job y configuraciones de reintentos \(p. 709\)](#)

Crear plantillas de trabajo personalizadas a partir de plantillas administradas

Puede utilizar unAWSplantilla administrada y un trabajo personalizado como punto de partida para crear su propia plantilla de trabajo personalizada. Para crear una plantilla de trabajo personalizada, cree primero un trabajo desde suAWSplantilla administrada tal y como se describe en la sección anterior.

A continuación, puede guardar el trabajo personalizado como plantilla para crear su propia plantilla de trabajo personalizada. Para guardar como plantilla:

1. Vaya a[Centro de Job delAWS IoTconsola](#) y elige el trabajo que contiene la plantilla administrada.

2. ElegirGuardar como plantilla de trabajo, a continuación, cree su plantilla de trabajo personalizada. Para obtener más información acerca de la creación de una plantilla de trabajo personalizada, consulte[Cree una plantilla de trabajo a partir de un trabajo existente \(p. 703\)](#).

Crear un trabajo deAWSplantillas gestionadas mediante elAWS CLI

UsarAWS CLIPara obtener información sobreAWSplantillas gestionadas y cree un trabajo utilizando estas plantillas. A continuación, puede guardar el trabajo como plantilla y, a continuación, crear su propia plantilla personalizada.

Enumera plantillas administradas

La[list-managed-job-templates](#) AWS CLI enumera todas las plantillas de trabajo deCuenta de AWS.

```
aws iot list-managed-job-templates
```

De forma predeterminada, al ejecutar este comando se muestran todos los disponiblesAWSplantillas gestionadas y sus detalles.

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

```
        "description": "A managed job template for restarting one or more system
services.",
        "environments": [
            "LINUX"
        ],
        "templateVersion": "1.0"
    ]
}
```

Para obtener más información, consulte[Plantillas de trabajo gestionadas por listas](#).

Obtener información detallada sobre una plantilla administrada

La[describe-managed-job-template](#) AWS CLI obtiene detalles acerca de una plantilla de trabajo especificada. Especifique el nombre de la plantilla de trabajo y una versión de plantilla opcional. Si no se especifica la versión de la plantilla, se devuelve la versión predefinida y predeterminada. A continuación se muestra un ejemplo de ejecución del comando para obtener detalles sobre el[AWS-Download-File](#) plantilla de.

```
aws iot describe-managed-job-template \
--template-name AWS-Download-File
```

El comando muestra los detalles de la plantilla y el ARN, su documento de trabajo y el[documentParameters](#) parámetro, que es una lista de pares de valores clave de parámetros de entrada de la plantilla. Para obtener información acerca de las distintas plantillas y parámetros de entrada, consulte[Acciones remotas de plantillas gestionadas y documentos de trabajo \(p. 690\)](#).

Note

La[documentParameters](#) objeto devuelto al utilizar esta API solo debe utilizarse al crear trabajos desde AWSplantillas administradas. El objeto no debe utilizarse para plantillas de trabajo personalizadas. Para ver un ejemplo que muestra cómo utilizar este parámetro, consulte[Crear un trabajo mediante plantillas administradas \(p. 701\)](#).

```
{
    "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": "User to run the job as on the device."}
```

```
        "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 obtener más información, consulte [Describir plantilla de trabajo administrada](#).

Crear un trabajo mediante plantillas administradas

La `create-job` AWS CLI se puede utilizar para crear un trabajo a partir de una plantilla de trabajo. Se dirige a un dispositivo llamado `thingOne` y especifica el nombre de recurso de Amazon (ARN) de la plantilla administrada que se usará como base para el trabajo. Puede anular las configuraciones avanzadas, como las configuraciones de tiempo de espera y cancelación, pasando los parámetros asociados de la `create-job` comando.

En el ejemplo se muestra cómo crear un trabajo que utilice la `AWS-Download-File` plantilla de. También muestra cómo especificar los parámetros de entrada de la plantilla mediante el `document-parameters` parámetro.

Note

Debe utilizar el `document-parameters` objeto solo con AWS plantillas administradas. Este objeto no debe utilizarse con plantillas de trabajo personalizadas.

```
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
```

donde:

- `region` es la región de AWS.
- `account-id` es el número de cuenta de AWS.
- `thingOne` es el nombre de la cosa de IoT a la que se dirige el trabajo.
- `AWS-Download-File:1.0` es el nombre de la plantilla administrada.
- `https://example.com/index.html` es la URL desde la que descargar el archivo.
- `https://path/to/file/index` es la ruta del dispositivo para almacenar el archivo descargado.

Al ejecutar este comando se crea un trabajo para la plantilla, `AWS-Download-File`, tal y como se muestra a continuación.

```
{  
    "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."  
}
```

Cree una plantilla de trabajo personalizada a partir de plantillas administradas

1. Cree un trabajo con una plantilla administrada tal y como se describe en la sección anterior.
2. Cree una plantilla de trabajo personalizada mediante el ARN del trabajo que ha creado. Para obtener más información, consulte [Cree una plantilla de trabajo a partir de un trabajo existente \(p. 705\)](#).

Creación de plantillas de trabajo personalizadas

Puede crear plantillas de trabajo mediante elAWS CLy laAWS IoTconsola de . También puede crear trabajos a partir de plantillas de trabajos mediante elAWS CLI, elAWS IoTconsola y Fleet Hub paraAWS IoTAplicaciones web de administración de dispositivos. Para obtener más información acerca de cómo trabajar con plantillas de trabajo en aplicaciones de Fleet Hub, consulte [Trabajo con plantillas de trabajo en Fleet Hub paraAWS IoTAdministración de dispositivos](#).

Temas

- [Cree plantillas de trabajo personalizadas mediante elAWS Management Console \(p. 702\)](#)
- [Cree plantillas de trabajo personalizadas mediante elAWS CLI \(p. 704\)](#)

Cree plantillas de trabajo personalizadas mediante elAWS Management Console

En este tema se explica cómo crear, eliminar y ver detalles acerca de las plantillas de trabajo mediante laAWS IoTconsola de .

Creación de una plantilla de trabajo personalizada

Puede crear un original plantilla de trabajo personalizada o crea una plantilla de trabajo a partir de un trabajo existente. También puede crear una plantilla de trabajo personalizada a partir de un trabajo existente que se creó con una AWSplantilla administrada. Para obtener más información, consulte [Crear plantillas de trabajo personalizadas a partir de plantillas administradas \(p. 698\)](#).

Crear una plantilla de trabajo original

1. Empieza a crear tu plantilla de trabajo
 1. Vaya a [Centro de plantillas de Job delAWS IoTconsola](#) y elige el Plantillas personalizadas Pestaña.
 2. Elegir Creación de plantillas de trabajo.

Note

También puede ir a la Plantillas de trabajo página de la Servicios relacionados página debajo Fleet Hub.

2. Especificar propiedades de plantilla de trabajo

En el navegador Creación de plantillas de trabajo, introduzca un identificador alfanumérico para el nombre de su puesto y una descripción alfanumérica para proporcionar detalles adicionales sobre la plantilla.

Note

No es recomendable utilizar datos personales en los ID de trabajos o en las descripciones.

3. Proporcionar documento de trabajo

Proporcione un archivo de trabajo JSON que se almacena en un bucket de S3 o como documento de trabajo en línea especificado en el trabajo. Este archivo de trabajo se convertirá en el documento de trabajo cuando cree un trabajo con esta plantilla.

Si el archivo de trabajo se almacena en un bucket de S3, introduzca la URL de S3 o seleccione **Examinar S3**, a continuación, navegue hasta el documento de trabajo y selecciónelo.

Note

Solo puede seleccionar buckets de S3 de la región actual.

4. Siga añadiendo configuraciones adicionales para su trabajo y, a continuación, revise y cree su trabajo.
Para obtener más información acerca de las configuraciones adicionales, consulte:

- [Despliegue de trabajos y configuraciones de anulaciones \(p. 708\)](#)
- [Tiempo de espera de ejecución de Job y configuraciones de reintentos \(p. 709\)](#)

Cree una plantilla de trabajo a partir de un trabajo existente

1. Elija su trabajo

1. Vaya a [Centro de Job del AWS IoT consola](#) y elija el trabajo que desea utilizar como base de su plantilla de trabajo.
2. Elegir Guardar como plantilla de trabajo.

Note

Opcionalmente, puede elegir un documento de trabajo diferente o editar las configuraciones avanzadas del trabajo original y, a continuación, elegir **Creación de plantillas de trabajo**. La nueva plantilla de trabajo aparece en la **Plantillas de trabajo** (Se ha creado el certificado).

2. Especificar propiedades de plantilla de trabajo

En el navegador **Creación de plantillas de trabajo**, introduzca un identificador alfanumérico para el nombre de su puesto y una descripción alfanumérica para proporcionar detalles adicionales sobre la plantilla.

Note

El documento de trabajo es el archivo de trabajo que especificó al crear la plantilla. Si el documento de trabajo se especifica dentro del trabajo en lugar de una ubicación de S3, puede verlo en la página de detalles de este trabajo.

3. Siga añadiendo configuraciones adicionales para su trabajo y, a continuación, revise y cree su trabajo.
Para obtener más información acerca de las configuraciones adicionales, consulte:

- [Despliegue de trabajos y configuraciones de anulaciones \(p. 708\)](#)
- [Tiempo de espera de ejecución de Job y configuraciones de reintentos \(p. 709\)](#)

Cree un trabajo a partir de una plantilla de trabajo personalizada

Puede crear un trabajo a partir de una plantilla de trabajo personalizada yendo a la página de detalles de la plantilla de trabajo tal y como se describe en este tema. También puede crear un trabajo o elegir la

plantilla de trabajo que desea utilizar al ejecutar el flujo de trabajo de creación de trabajos. Para obtener más información, consulte [Cree y administre trabajos de mediante elAWS Management Console \(p. 679\)](#).

En este tema se muestra cómo crear un trabajo desde la página de detalles de una plantilla de trabajo personalizada. También puede crear un trabajo a partir de unAWSplantilla administrada. Para obtener más información, consulte [Crear un trabajo mediante plantillas administradas \(p. 698\)](#).

1. Elija su plantilla de trabajo personalizada

Vaya a[Centro de plantillas de Job delAWS IoTconsolay](#) elige elPlantillas personalizadasy, a continuación, elija su plantilla.

2. Crear un trabajo utilizando una plantilla personalizada

Para crear un trabajo:

1. En la página de detalles de la plantilla, eligeCrear el trabajo.

La consola cambia alPropiedades de trabajo personalizadaspaso delCrear el trabajoflujo de trabajo en el que se ha añadido la configuración de la plantilla.

2. Introduzca un nombre de trabajo alfanumérico único y una descripción y etiquetas opcionales y, a continuación, elijaPróximo.

3. Elija las cosas o grupos de cosas como objetivos de trabajo que desea ejecutar en este trabajo.

En el navegadordocumento de Job, la plantilla se muestra con sus ajustes de configuración. Si desea utilizar otro documento de trabajo, elijaNavegary selecciona otro depósito y documento. Elija Next (Siguiente).

4. En la páginaConfiguración de Job, elija el tipo de trabajo comocontinuo o un trabajo de instantánea. Un trabajo de instantáneas se completa cuando finaliza su ejecución en los dispositivos y grupos de destino. Un trabajo continuo se aplica a los grupos de cosas y se ejecuta en cualquier dispositivo que agregue a un grupo de destino especificado.

5. Siga añadiendo configuraciones adicionales para su trabajo y, a continuación, revise y cree su trabajo. Para obtener más información acerca de las configuraciones adicionales, consulte:

- [Despliegue de trabajos y configuraciones de anulaciones \(p. 708\)](#)
- [Tiempo de espera de ejecución de Job y configuraciones de reintentos \(p. 709\)](#)

También puede crear trabajos a partir de plantillas de trabajos con aplicaciones web de Fleet Hub. Para obtener más información acerca de la creación de trabajos en Fleet Hub, consulte[Trabajo con plantillas de trabajo en Fleet Hub paraAWS IoTAdministración de dispositivos](#).

Eliminación de una plantilla de trabajo

Para eliminar una plantilla de trabajo, vaya primero a la[Centro de plantillas de Job delAWS IoTconsolay](#) elige elPlantillas personalizadasPestaña. A continuación, elija la plantilla de trabajo que deseé eliminar y elijaBorrar.

Note

Una eliminación es permanente y la plantilla de trabajo ya no aparece en elPlantillas personalizadasPestaña.

Cree plantillas de trabajo personalizadas mediante elAWS CLI

En este tema se explica cómo crear, eliminar y recuperar detalles acerca de las plantillas de trabajo mediante laAWS CLI.

Creación de una plantilla de trabajo desde cero

Los siguientes ejemplos de AWS CLI muestran cómo crear un trabajo usando un documento de trabajo ([job-document.json](#)) almacenado en un bucket de S3 y un rol con permiso para descargar archivos de Amazon S3 ([Función de descarga de S3](#)).

```
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}"
```

El parámetro `timeout-config` opcional especifica la cantidad de tiempo que cada dispositivo tiene para finalizar su ejecución del trabajo. El temporizador comienza cuando el estado de ejecución del trabajo se establece en `IN_PROGRESS`. Si el estado de ejecución del trabajo no se establece en otro estado terminal antes de que se cumpla el plazo, se establecerá `TIMED_OUT`.

El temporizador en curso no se puede actualizar y se aplica a todas las ejecuciones de trabajos para el trabajo. Cuando la ejecución de un trabajo permanece en estado `IN_PROGRESS` durante un periodo superior a este intervalo, la ejecución del trabajo producirá un error y cambiará al estado final `TIMED_OUT`. AWS IoT también publica una notificación MQTT.

Para obtener más información acerca de la creación de configuraciones sobre despliegues de trabajos y anulaciones, consulte [Despliegue de trabajos y configuración de anulaciones](#).

Note

Los documentos de Jobs que se especifican como archivos de Amazon S3 se recuperan en el momento en el que crea el Job. Si cambia el contenido del archivo de Amazon S3 que usó como origen de su documento de trabajo después de haber creado el trabajo, lo que se envía a los destinos del trabajo no cambia.

Cree una plantilla de trabajo a partir de un trabajo existente

Los siguientes ejemplos de AWS CLI crean una plantilla de trabajo especificando el nombre de recurso de Amazon (ARN) de un trabajo existente. La nueva plantilla de trabajo utiliza todas las configuraciones especificadas en el trabajo. Opcionalmente, puede cambiar cualquiera de las configuraciones del trabajo existente utilizando cualquiera de los parámetros opcionales.

```
aws iot create-job-template \
    --job-arn arn:aws:iot:<region>:123456789012:job/<job-name> \
    --timeout-config inProgressTimeoutInMinutes=100
```

Obtener información detallada sobre una plantilla de trabajo

Los siguientes ejemplos de AWS CLI obtienen detalles acerca de una plantilla de trabajo especificada.

```
aws iot describe-job-template \
```

```
--job-template-id template-id
```

El comando muestra el resultado siguiente.

```
{  
    "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  
    }  
}
```

Enumera plantillas de trabajo

Los siguientes ejemplos de AWS CLI enumeran todas las plantillas de trabajo de la cuenta de AWS.

```
aws iot list-job-templates
```

El comando muestra el resultado siguiente.

```
{  
    "jobTemplates": [  
        {  
            "createdAt": number,  
            "description": "string",  
            "jobTemplateArn": "string",  
            "jobTemplateId": "string"  
        }  
    ]  
}
```

```
        },
    ],
    "nextToken": "string"
}
```

Para recuperar páginas adicionales de resultados, utilice el valor de la `nextToken`.

Eliminación de una plantilla de trabajo

Los siguientes ejemplos de AWS CLI eliminan una plantilla de trabajo especificada.

```
aws iot delete-job-template \
--job-template-id template-id
```

El comando no muestra ninguna salida.

Cree un trabajo a partir de una plantilla de trabajo personalizada

Los siguientes ejemplos de AWS CLI crean un trabajo a partir de una plantilla de trabajo personalizada. Se dirige a un dispositivo llamado `thingOne` y especifica el nombre de recurso de Amazon (ARN) de la plantilla de trabajo que se va a utilizar como base para el trabajo. Puede anular las configuraciones avanzadas, como las configuraciones de tiempo de espera y cancelación, pasando los parámetros asociados de la `create-job` comando.

Warning

La `document-parameters` debe utilizarse con el objeto `create-jobs` solo cuando se crean trabajos desde AWS plantillas administradas. Este objeto no debe utilizarse con plantillas de trabajo personalizadas. Para ver un ejemplo que muestra cómo crear trabajos con este parámetro, consulte [Crear un trabajo mediante plantillas administradas \(p. 701\)](#).

```
aws iot create-job \
--targets arn:aws:iot:region:123456789012:thing/thingOne \
--job-template-arn arn:aws:iot:region:123456789012:jobtemplate/template-id
```

Trabajoconfigurations

Puede tener las siguientes configuraciones adicionales para cada trabajo que despliegue en los destinos especificados.

- **despliegue:** Esta configuración define cuántos dispositivos reciben el documento de trabajo cada minuto.
- **Abortar:** Utilice esta configuración para cancelar un trabajo en casos tales como cuando algunos dispositivos no reciben la notificación de trabajo o los dispositivos informan de errores en sus ejecuciones de trabajos.
- **Timeout (Tiempo de espera):** Si no hay respuesta de tus objetivos de trabajo dentro de un plazo determinado después de que se hayan iniciado las ejecuciones de sus trabajos, el trabajo puede fallar.
- **Intentar de nuevo:** Si el dispositivo informa de un error al intentar completar la ejecución de un trabajo o si se agota el tiempo de ejecución del trabajo, puede utilizar esta configuración para volver a intentar la ejecución del trabajo del dispositivo.

Mediante estas configuraciones, puede supervisar el estado de la ejecución de su trabajo y evitar que se envíe una actualización incorrecta a toda una flota.

Temas

- [Cómo funcionan las configuraciones de trabajos \(p. 708\)](#)
- [Especifica configuraciones adicionales \(p. 713\)](#)

Cómo funcionan las configuraciones de trabajos

Se utilizan las configuraciones de implementación y cancelación cuando implementa un trabajo y las configuraciones de tiempo de espera y reintento para la ejecución de trabajos. En las secciones siguientes se muestra más información acerca de cómo funcionan estas configuraciones.

Temas

- [Despliegue de trabajos y configuraciones de anulaciones \(p. 708\)](#)
- [Tiempo de espera de ejecución de Job y configuraciones de reintentos \(p. 709\)](#)

Despliegue de trabajos y configuraciones de anulaciones

Puede utilizar las configuraciones de implementación y cancelación de trabajos para definir cuántos dispositivos reciben el documento de trabajo cada minuto, así como los criterios para cancelar un trabajo cuando un determinado número de dispositivos no recibe un documento de trabajo.

Configuración de implementación de Job

Puede especificar la rapidez con la que se notifica a los destinos la ejecución de un trabajo pendiente. También puede crear una implementación por etapas para administrar mejor las actualizaciones, los reinicios y otras operaciones. Para especificar cómo se notifican a los objetivos, utilice las tasas de despliegue de trabajos.

Tasas de despliegue de Job

Puede crear una configuración de despliegue utilizando una tasa de despliegue constante o una velocidad de despliegue exponencial. Para especificar el número máximo de objetivos de trabajos para informar por minuto, utilice una tasa de despliegue constante.

AWS IoT trabajos se pueden implementar utilizando tasas de despliegue exponencial según se cumplan distintos criterios y umbrales. Si el número de trabajos con error coincide con un conjunto de criterios especificados, puede cancelar la implementación del trabajo. Se establecen los criterios de la tasa de despliegue de trabajos cuando crea un trabajo mediante el `JobExecutionsRolloutConfig` objeto. También se establecen los criterios de cancelación de trabajos en la creación del trabajo mediante el `AbortConfig` objeto.

El siguiente ejemplo muestra cómo funcionan las tasas de despliegue. Por ejemplo, una implementación de trabajos con una tasa base de 50 por minuto, factor de incremento de 2 y número de dispositivos notificados y exitosos como 1000, funcionaría de la siguiente manera: El trabajo comenzará con una tasa de 50 ejecuciones de trabajos por minuto y continuará según esa tasa hasta que bien 1000 objetos hayan recibido notificaciones de ejecución de trabajos o se hayan producido correctamente 1000 ejecuciones de trabajos.

En la siguiente tabla se muestra cómo se produciría el despliegue durante los primeros cuatro incrementos.

Tasa de despliegue por minuto	50	100	200	400
Número de dispositivos notificados o ejecuciones de trabajo correctas para satisfacer un aumento de tasas	1 000	2000	3 000	4000

Tasas de implementación de Job para trabajos continuos utilizando grupos de cosas dinámicos

Cuando utiliza un trabajo continuo para implementar operaciones remotas en su flota, AWS IoT crea ejecuciones de trabajos para dispositivos del grupo de destino. Estas ejecuciones de trabajos siguen desplegándose en cualquier dispositivo nuevo que se agregue al grupo, incluso después de que se haya creado el trabajo. La configuración de implantación puede controlar las tasas de despliegue solo para los dispositivos que se agregan al grupo hasta la creación del trabajo. Después de crear un trabajo, para cualquier dispositivo nuevo, las ejecuciones de trabajos se crean casi en tiempo real tan pronto como los dispositivos se unen al grupo de destino.

Configuración de cancelación de Job

Utilice esta configuración para crear un criterio para cancelar un trabajo cuando un porcentaje umbral de dispositivos cumple esos criterios. Por ejemplo, puede utilizar esta configuración para cancelar un trabajo en los siguientes casos:

- Cuando un porcentaje umbral de tus dispositivos no recibe las notificaciones de ejecución del trabajo, por ejemplo, cuando el dispositivo es incompatible para una actualización de OTA. En este caso, el dispositivo puede informar de `UNREJECTED` Estado.
- Cuando un porcentaje umbral de dispositivos informa de errores en sus ejecuciones de trabajos, por ejemplo, cuando el dispositivo encuentra una desconexión al intentar descargar el documento de trabajo desde una URL de S3. En tales casos, el dispositivo debe estar programado para informar `FAILURE` Estado a AWS IoT.
- Cuando `TIMED_OUT` el estado se informa a medida que se agota el tiempo de ejecución del trabajo para un porcentaje umbral de dispositivos después de que se hayan iniciado las ejecuciones del trabajo.
- Cuando hay varios errores de reinicio. Al agregar una configuración de reinicio, cada intento de reinicio puede generar cargos adicionales a su Cuenta de AWS. En tales casos, la cancelación del trabajo puede cancelar las ejecuciones de trabajos en cola y evitar intentos de reinicio de estas ejecuciones. Para obtener más información acerca de la configuración de reinicio y su uso con la configuración anular, consulte [Tiempo de espera de ejecución de Job y configuraciones de reinicio \(p. 709\)](#).

Puede configurar una condición de anulación de un trabajo mediante la AWS IoT consola o la AWS IoT API de trabajos.

Tiempo de espera de ejecución de Job y configuraciones de reinicio

Utilice la configuración de tiempo de espera de ejecución de trabajos para enviarle [Notificaciones de trabajos \(p. 724\)](#) cuando la ejecución de un trabajo ha estado en curso durante más tiempo que la duración establecida. Utilice la configuración de reinicio de ejecución de trabajos para volver a intentar la ejecución cuando el trabajo falla o se agota el tiempo de espera.

Configuración de tiempo de espera de ejecución de Job

Utilice la configuración de tiempo de espera de ejecución de trabajos para notificarle cada vez que la ejecución de un trabajo se queda atascada en el `IN_PROGRESS` estado durante un período de tiempo inesperadamente largo. Cuando el trabajo es `IN_PROGRESS`, puede monitorizar el progreso de la ejecución de su trabajo.

Temporizadores para tiempos de espera de trabajo

Existen dos tipos de temporizadores: temporizadores en curso y temporizadores de pasos.

Temporizadores en curso

Cuando creas un trabajo o una plantilla de trabajo, puedes especificar un valor para el temporizador en curso que oscila entre 1 minuto y 7 días. Puede actualizar el valor de este temporizador hasta el inicio de la ejecución del trabajo. Una vez iniciado el temporizador, no se puede actualizar y el valor del temporizador se aplica a todas las ejecuciones de trabajos para el trabajo. Cuando la ejecución de un trabajo permanece en estado `IN_PROGRESS` durante un periodo superior a este intervalo, la ejecución del trabajo producirá un error y cambiará al estado final `TIMED_OUT`. AWS IoT también publica una notificación MQTT.

Temporizador paso

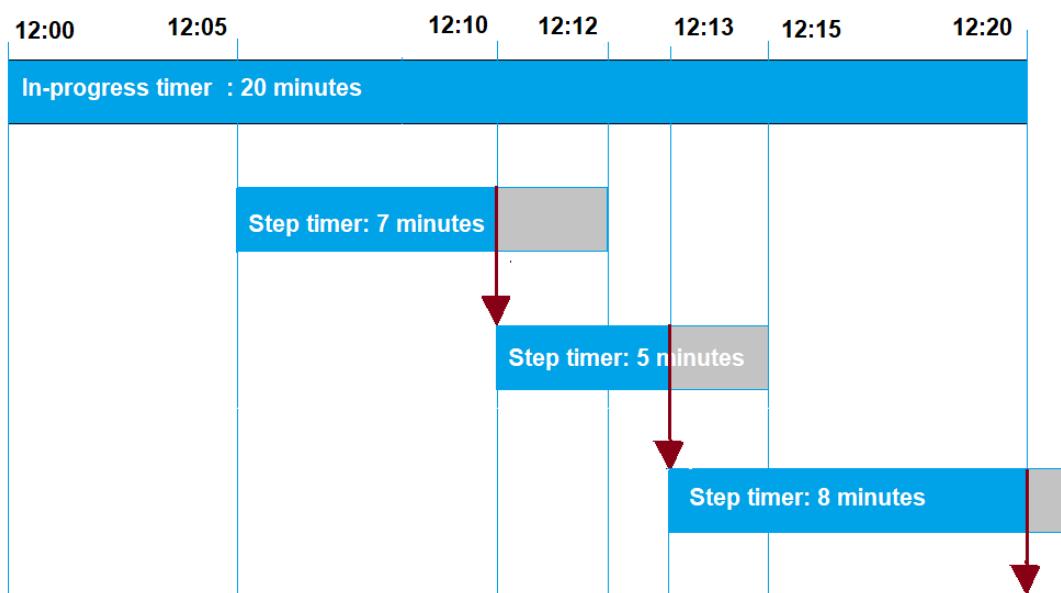
También puede configurar un temporizador de pasos que se aplique únicamente a la ejecución de trabajos que deseé actualizar. Este temporizador no tiene ningún efecto en el temporizador en curso. Cada vez que actualice la ejecución de un trabajo, puede establecer un nuevo valor para el temporizador de pasos. También puede crear un temporizador de pasos nuevo cuando inicie la siguiente ejecución de trabajos pendientes de un objeto. Si la ejecución del trabajo permanece en estado `IN_PROGRESS` durante un periodo superior a este intervalo del temporizador de pasos, generará un error y cambiará al estado terminal `TIMED_OUT`.

Note

Puede configurar el temporizador en curso utilizando el AWS IoT consola de oAWS IoT API de trabajos. Para especificar el temporizador de pasos, utilice la API.

Cómo funcionan los temporizadores para los tiempos de espera del trabajo

A continuación se ilustran las formas en las que los tiempos de espera en curso y por pasos interactúan entre sí en un período de tiempo de espera de 20 minutos.



A continuación se muestran los distintos pasos:

1. 12:00

Se crea un nuevo trabajo y se inicia un temporizador en curso durante veinte minutos al crear un trabajo. El temporizador en curso se pone en funcionamiento y la ejecución del trabajo cambia a `IN_PROGRESS` Estado.

2. 12:05. P. M.

Se crea un temporizador de pasos nuevo con un valor de 7 minutos. El tiempo de ejecución del trabajo se agotará a las 12:12 PM.

3. 12:10. P. M.

Se crea un temporizador de pasos nuevo con un valor de 5 minutos. Cuando se crea un temporizador de pasos nuevo, se descarta el temporizador de pasos anterior y el tiempo de espera de la ejecución del trabajo a las 12:15 PM (12:15 h).

4. 12:13. P. M.

Se crea un temporizador de pasos nuevo con un valor de 9 minutos. El temporizador de paso anterior se descarta y la ejecución del trabajo se agotará a las 12:20 p.m., porque el temporizador en curso se agota a las 12:20 PM. El temporizador de pasos no puede superar el límite absoluto del temporizador en curso.

Configuración de reinicio de ejecución de Job

Puede utilizar la configuración de reinicio para volver a intentar la ejecución del trabajo cuando se cumple un determinado conjunto de criterios. Se puede intentar reintentar cuando se agota el tiempo de espera de un trabajo o cuando el dispositivo falla. Para volver a intentar la ejecución debido a un error de tiempo de espera, debe habilitar la configuración de tiempo de espera.

Cómo utilizar la configuración de reinicios

A continuación se muestra cómo utilizar la configuración de reinicio:

1. Determine si se va a utilizar la configuración de reinicios para FAILED, TIMED_OUT, o ambos criterios de fallo. Para el registro TIMED_OUT status, una vez que se haya notificado el estado, AWS IoT Los trabajos reintentan automáticamente la ejecución del trabajo para el dispositivo.
2. Para el registro FAILED status, compruebe si el error de ejecución del trabajo se puede volver a intentar. Si se puede volver a intentar, programa el dispositivo para que informe de un FAILURE Estado a AWS IoT. En la siguiente sección se describe más información sobre los fallos reprobables y no reprobables.
3. Especifique el número de reinicios que se utilizarán para cada tipo de error mediante la información anterior. En un único dispositivo, puede especificar hasta 10 reinicios para ambos tipos de error combinados. Los intentos de reinicio se detienen automáticamente cuando una ejecución se realiza correctamente o cuando alcanza el número especificado de intentos.
4. Agregue una configuración de cancelación para cancelar el trabajo en caso de fallos de reinicios repetidos, evitando así que se incurran en cargos adicionales en caso de que haya un gran número de intentos de reinicio.

Reintentar y anular configuración de anulaciones

Cada intento de reinicio incurrirá en cargos adicionales a tu Cuenta de AWS. Para evitar que se incurran en cargos adicionales por fallos de reinicios repetidos, recomendamos agregar una configuración de cancelación. Para obtener más información acerca de los precios, consulte [Precios de AWS IoT Device Management](#).

Es posible que se produzcan varios errores de reinicio cuando un porcentaje de umbral elevado de los dispositivos se agota el tiempo de espera o informa de un fallo. En este caso, puede utilizar la configuración de cancelación para cancelar el trabajo y evitar cualquier ejecución de trabajos en cola o intentos de reinicios adicionales.

Note

Cuando se cumplen los criterios de cancelación para cancelar la ejecución de un trabajo, solo QUEUED se cancelan las ejecuciones de trabajos. No se intentarán reinicios en cola para el dispositivo. Sin embargo, las ejecuciones de trabajo actuales que tienen un IN_PROGRESS estado no se cancelarán.

Antes de volver a intentar una ejecución de trabajo fallida, también le recomendamos que compruebe si el error de ejecución del trabajo se puede volver a intentar, tal y como se describe en la siguiente sección.

Reintento para el tipo de fallo de**FAILED**

Para intentar reintentos para el tipo de error de**FAILED**, los dispositivos deben estar programados para informar de la**FAILURE**estado de una ejecución de trabajo fallida enAWS IoT. También debe establecer la configuración de reinicio con los criterios para volver a intentarlo.**FAILEDejecuciones de trabajos** y especifique el número de reintentos que se van a realizar. CuandoAWS IoTJobs detecta el estado **FAILURE** y, a continuación, intentará reiniciar automáticamente la ejecución del trabajo del dispositivo. Los reintentos continúan hasta que la ejecución del trabajo se realiza correctamente o cuando alcanza el número máximo de intentos de reinicio.

Puede realizar un seguimiento de cada intento de reinicio y del trabajo que se ejecuta en estos dispositivos. Si realiza el seguimiento del estado de ejecución, una vez que se haya intentado el número especificado de reintentos, puede utilizar el dispositivo para informar de errores e iniciar otro intento de reinicio.

Fallas reintentables y no reprobables

El error en la ejecución del trabajo puede ser reintentable o no recuperable. Cada intento de reinicio puede ocurrir en cargos a tuCuenta de AWS. Para evitar ocurrir en cargos adicionales por varios intentos de reinicio, primero considere comprobar si el error de ejecución del trabajo se puede volver a intentar. Un ejemplo de error reintentable incluye un error de conexión que el dispositivo detecta al intentar descargar el documento de trabajo desde una URL de S3. Si el error de ejecución del trabajo se puede volver a intentar, puede programar el dispositivo para que informe de un**FAILURE**estado en caso de que se produzca un error en la ejecución del trabajo y establezca la configuración de reinicio para volver a intentarlo**FAILEDejecuciones**.

Si no se puede volver a intentar la ejecución, para evitar volver a intentar la ejecución del trabajo y, potencialmente, ocurrir en cargos adicionales en su cuenta, le recomendamos que programa el dispositivo para que informe de un**REJECTED**Estado aAWS IoT. Algunos ejemplos de errores no reprobables incluyen cuando el dispositivo es incompatible con recibir una actualización de trabajo o cuando experimenta un error de memoria al ejecutar un trabajo. En estos casos,AWS IoTLos trabajos no volverán a intentar la ejecución del trabajo porque vuelve a intentar la ejecución del trabajo solo cuando detecta un**FAILEDOTIMED_OUT**Estado.

Una vez que haya determinado que un error en la ejecución de un trabajo se puede volver a intentar, si un intento de reinicio sigue fallando, considere comprobar los registros del dispositivo.

Reintento para el tipo de fallo de**TIMEOUT**

Si habilita el tiempo de espera al crear un trabajo, entoncesAWS IoTLos trabajos intentarán volver a intentar la ejecución del trabajo del dispositivo cuando el estado cambie de**IN_PROGRESS**a**TIMED_OUT**. Este cambio de estado puede producirse cuando se agota el tiempo de espera del temporizador en curso o cuando se encuentra un temporizador de pasos que especifique**IN_PROGRESS**y luego se agota el tiempo. Los reintentos continúan hasta que la ejecución del trabajo se realiza correctamente o cuando alcanza el número máximo de intentos de reintentos para este tipo de error.

Actualizaciones continuas de trabajos y miembros de grupos de cosas

Para trabajos continuos que tienen el estado del trabajo como**IN_PROGRESS**, el número de intentos de reinicio se restablece a cero cuando hay actualizaciones de la pertenencia a un grupo de una cosa. Por ejemplo, tenga en cuenta que ha especificado cinco intentos de reinicio y tres reintentos ya se han realizado. Si ahora se elimina una cosa del grupo de cosas y luego se vuelve a unir al grupo, como en el caso de los grupos de cosas dinámicos, el número de intentos de reinicio se restablece a cero. Ahora puedes realizar cinco intentos de reinicio para tu grupo de cosas en lugar de los dos intentos que quedaban. Además, cuando se elimina una cosa del grupo de cosas, se cancelan los intentos de reinicio adicionales.

Especifica configuraciones adicionales

Al crear un trabajo o una plantillas de trabajo, puede especificar estas configuraciones adicionales. A continuación se muestra cuándo puede especificar estas configuraciones.

- Al crear una plantillas de trabajo personalizada. La configuración adicional que especifique se guardará cuando cree un trabajo a partir de la plantilla.
- Al crear un trabajo personalizado mediante un archivo de trabajo. El archivo de trabajo puede ser un archivo JSON que se ha cargado en un bucket de S3.
- Al crear un trabajo personalizado mediante una plantilla de trabajo personalizada. Si la plantilla ya tiene especificada esta configuración, puede reutilizarla o anularla especificando nuevos ajustes de configuración.
- Al crear un trabajo personalizado mediante unAWSplantilla administrada.

Temas

- [Especificar configuraciones de trabajo mediante elAWS Management Console \(p. 713\)](#)
- [Especificar configuraciones de trabajo mediante elAWS IoTAPI de trabajos \(p. 715\)](#)

Especificar configuraciones de trabajo mediante elAWS Management Console

Puede agregar las distintas configuraciones de su trabajo mediante laAWS IoTconsola de . Después de crear un trabajo, puedes ver los detalles de estado de las configuraciones de trabajo en la página de detalles del trabajo. Para obtener más información acerca de las distintas configuraciones y cómo funcionan, consulte[Cómo funcionan las configuraciones de trabajos \(p. 708\)](#).

Agregue las configuraciones de trabajo al crear un trabajo o una plantilla de trabajo.

Al crear una plantillas de trabajo personalizada

Para especificar la configuración de implantación al crear una plantilla de trabajo personalizada

1. Vaya a[Centro de plantillas de Job delAWS IoTconsolay eligeCreación de plantillas de trabajos.](#)
2. Especifique las propiedades de la plantilla de trabajo, proporcione el documento de trabajo, amplíe la configuración que desea agregar y, a continuación, especifique los parámetros de configuración.

Al crear un trabajo personalizado

Para especificar la configuración de implantación al crear un trabajo personalizado

1. Vaya a[Centro de Job delAWS IoTconsolay eligeCrear el trabajo.](#)
2. ElegirCreación de un trabajo personalizadoy especifique las propiedades del trabajo, los destinos y si se va a utilizar un archivo de trabajo o una plantilla para el documento de trabajo. Puede utilizar una plantilla personalizada o unAWSplantilla administrada.
3. Seleccione la configuración del trabajo y, a continuación, expandaConfiguración de desplieguepara especificar si desea utilizar unTasa constanteoTasa exponencial. A continuación, especifique los parámetros de configuración.

En la siguiente sección se muestran los parámetros que puede especificar para cada configuración.

Configuración de despliegue

Puede especificar si desea utilizar una tasa de despliegue constante o una tasa exponencial.

- Establecer una velocidad de despliegue constante

Para establecer una tasa constante para las ejecuciones de trabajos, elijaTasa constante, a continuación, especificar laMáximo por minutoPara el límite superior de la tasa. Este valor es opcional y va de 1 a 1000. Si no lo configuras, utiliza 1000 como valor predeterminado.

- Establecer una tasa de despliegue exponencial

Para establecer una tasa exponencial, elijaTasa exponencial, a continuación, especifique estos parámetros:

- Tasa base por minuto

Tasa según la cual se ejecutan los trabajos hasta laNúmero de dispositivos notificados oNúmero de dispositivos con éxito se cumple el umbral paraCriterios de aumento de.

- Factor de incremento

El factor exponencial según el cual aumenta la tasa de despliegue tras laNúmero de dispositivos notificados oNúmero de dispositivos con éxito se cumple el umbral paraCriterios de aumento de.

- Criterios de aumento de

El umbral para cualquiera de los dosNúmero de dispositivos notificados oNúmero de dispositivos con éxito.

Configuración Abort

ElegirAñadir nueva configuración dey especifique los siguientes parámetros para cada configuración:

- Tipo de error

Especifica los tipos de error que inician la anulación de un trabajo. Entre ellas se incluyenERROR,RECHAZADO,TIMED_OUT, o bienTODOS.

- Factor de incremento

Especifica el número de ejecuciones de trabajos completadas que deben ocurrir antes de que se cumplan los criterios de anulación del trabajo.

- Porcentaje de umbral

Especifica el número total de objetos ejecutados que inician la anulación de un trabajo.

Configuración de tiempo de espera

De forma predeterminada, no hay tiempo de espera y tu trabajo se ejecuta cancelado o eliminado. Para utilizar los tiempos de espera, elijaActivar timeout, a continuación, especifique un valor de espera comprendido entre 1 minuto y 7 días.

Configuración de reintento

Note

Una vez que se ha creado el trabajo, no podrá actualizar el número de reintentos. Solo puede quitar la configuración de reintentos para todos los tipos de fallos. Cuando cree un trabajo, tenga en cuenta el número adecuado de reintentos para utilizar para la configuración. Para evitar incurrir en costes excesivos debido a posibles fallos de reintentos, agregue una configuración de cancelación.

ElegirAñadir nueva configuración dey especifique los siguientes parámetros para cada configuración:

- Tipo de error

Especifica los tipos de error que deben activar un reintento de ejecución de trabajos. Entre ellas se incluyen Failed (Error), Timeout (Tiempo de espera), y Todos.

- Number of retries (Número de reintentos)

Especifica el número de reintentos para el elegidoTipo de error. Para ambos tipos de fallos combinados, se pueden intentar hasta 10 reintentos.

Especificando configuraciones de trabajo mediante elAWS IoT API de trabajos

Puede utilizar el[CreateJob](#) o el[CreateJobTemplate](#) API para especificar las distintas configuraciones de trabajo. A continuación se muestra cómo agregar estas configuraciones. Una vez añadidas las configuraciones, puede utilizar [JobExecutionSummary](#) o [JobExecutionSummaryForJob](#) para ver su estado.

Para obtener más información acerca de las distintas configuraciones y cómo funciona, consulte [Cómo funcionan las configuraciones de trabajos \(p. 708\)](#).

Configuración de despliegue

Puede especificar una tasa de despliegue constante o una tasa de despliegue exponencial para la configuración de despliegue.

- Establecer una velocidad de despliegue constante

Para establecer una velocidad de despliegue constante, utilice la [JobExecutionsRolloutConfig](#) objeto para agregar el `maximumPerMinute` parámetro de `CreateJobRequest`. Este parámetro especifica el límite superior de la tasa según la cual pueden producirse las ejecuciones de los trabajos. Este valor es opcional y va de 1 a 1000. Si no establece el valor, utiliza 1000 como valor predeterminado.

```
"jobExecutionsRolloutConfig": {  
    "maximumPerMinute": 1000  
}
```

- Establecer una tasa de despliegue exponencial

Para establecer una tasa de despliegue de trabajos variable, utilice la [JobExecutionsRolloutConfig](#) objeto. Puede configurar el `ExponentialRolloutRate` propiedad cuando ejecuta el `CreateJob` operación de la API. En el siguiente de ejemplo se establece una tasa de despliegue exponencial mediante la `exponentialRate` parámetro. Para obtener más información acerca de los parámetros, consulte [ExponentialRolloutRate](#)

```
{  
  ...  
  "jobExecutionsRolloutConfig": {  
    "exponentialRate": {  
      "baseRatePerMinute": 50,  
      "incrementFactor": 2,  
      "rateIncreaseCriteria": {  
        "numberOfNotifiedThings": 1000,  
        "numberOfSucceededThings": 1000  
      },  
      "maximumPerMinute": 1000  
    }  
  }  
  ...  
}
```

Dónde está el parámetro:

baseRatePerMinute

Especifica la tasa según la cual se ejecutan los trabajos hasta
la `numberOfNotifiedThings` o `number of SucceededThings` ha cumplido el umbral.

incrementFactor

Especifica el factor exponencial según el cual aumenta la tasa de despliegue tras
la `numberOfNotifiedThings` o `number of SucceededThings` ha cumplido el umbral.

rateIncreaseCriteria

Especifica el `numberOfNotifiedThings` o `number of SucceededThings` umbral.

Configuración Abort

Para agregar esta configuración mediante la API, especifique la `AbortConfig` cuando ejecuta el parámetro `createJob` o la `CreateJobTemplate` Operación de la API. En el siguiente ejemplo se muestra una configuración anular para un despliegue de trabajos que estaba experimentando varias ejecuciones fallidas, según se especifica con la `CreateJob` Operación de la API.

Note

Eliminar la ejecución de un trabajo afecta al valor de cálculo de la ejecución total completada. Cuando se anula un trabajo, el servicio crea valores `comment` y `reasonCode` automáticos para diferenciar una cancelación promovida por un usuario de una de anulación de un trabajo.

```
"abortConfig": {  
    "criteriaList": [  
        {  
            "action": "CANCEL",  
            "failureType": "FAILED",  
            "minNumberOfExecutedThings": 100,  
            "thresholdPercentage": 20  
        },  
        {  
            "action": "CANCEL",  
            "failureType": "TIMED_OUT",  
            "minNumberOfExecutedThings": 200,  
            "thresholdPercentage": 50  
        }  
    ]  
}
```

Dónde está el parámetro:

action

Especifica la acción que se debe realizar cuando se han cumplido los criterios de anulación. Este parámetro es necesario, y `CANCEL` es el único valor válido.

failureType

Especifica qué tipos de error deben iniciar la anulación de un trabajo. Los valores válidos son `FAILED`, `REJECTED`, `TIMED_OUT` y `ALL`.

minNumberOfExecutedThings

Especifica el número de ejecuciones de trabajos completadas que deben ocurrir antes de que se cumplan los criterios de anulación del trabajo. En este ejemplo, AWS IoT no comprueba si se debe anular un trabajo hasta que al menos 100 dispositivos hayan completado ejecuciones de trabajos.

thresholdPercentage

Especifica el número total de objetos para los que se ejecutan trabajos que pueden iniciar la anulación de un trabajo. En este ejemplo, AWS IoT comprueba secuencialmente e inicia un aborto de trabajo si se cumple el porcentaje de umbral. Si al menos el 20% de las ejecuciones completas fallaron después de que se hayan completado 100 ejecuciones, cancela la implementación del trabajo. Si no se cumplen estos criterios, AWS IoT comprueba si al menos el 50% de las ejecuciones finalizadas se agotaron después de que se hayan completado 200 ejecuciones. Si este es el caso, cancela la implementación del trabajo.

Configuración de tiempo de espera

Para agregar esta configuración mediante la API, especifique la [TimeoutConfig](#) cuando ejecuta el parámetro [createJob](#) o el [CreateJobTemplate](#) Operación de la API.

Para utilizar la configuración de tiempo de espera

1. Para configurar el temporizador en curso al crear un trabajo o una plantilla de trabajo, establezca un valor para el [inProgressTimeoutInMinutes](#) propiedad del opcional [TimeoutConfig](#) objeto.

```
"timeoutConfig": {  
    "inProgressTimeoutInMinutes": number  
}
```

2. Para especificar un temporizador de pasos para la ejecución de un trabajo, establezca un valor para [stepTimeoutInMinutes](#) cuando llamas [UpdateJobExecution](#). El temporizador de pasos se aplica únicamente a la ejecución del trabajo que actualice. Puede establecer un nuevo valor para este temporizador cada vez que actualice la ejecución de un trabajo.

Note

[UpdateJobExecution](#) también puede descartar un temporizador de pasos que ya se ha creado mediante la creación de un nuevo temporizador de pasos con un valor de -1.

```
{  
  ...  
  "statusDetails": {  
    "string" : "string"  
  },  
  "stepTimeoutInMinutes": number  
}
```

3. Para crear un nuevo temporizador de pasos, también puedes llamar a la [StartNextPendingJobExecution](#) Operación de la API.

Configuración de reintento

Note

Cuando cree un trabajo, tenga en cuenta el número adecuado de reintentos para utilizar para la configuración. Para evitar incurrir en costes excesivos debido a posibles fallos de reintento, agregue una configuración de cancelación. Una vez que se ha creado el trabajo, no podrá actualizar el número de reintentos. Solo puede establecer el número de reintentos en 0 mediante la [UpdateJob](#) Operación de la API.

Para agregar esta configuración mediante la API, especifique la [jobExecutionsRetryConfig](#) cuando ejecuta el parámetro [createJob](#) o el [CreateJobTemplate](#) Operación de la API.

```
{
```

```
...  
    "jobExecutionsRetryConfig": {  
        "criteriaList": [  
            {  
                "failureType": "string",  
                "numberOfRetries": number  
            }  
        ]  
    }  
...  
}
```

Donde `criteriaList`es una matriz que especifica la lista de criterios que determina el número de reintentos permitidos para cada tipo de error de un trabajo.

Dispositivos y trabajos

Los dispositivos se pueden comunicar conAWS IoTTrabajos que utilizan MQTT, firma HTTP versión 4 o HTTP TLS. Para determinar el punto final que se va a utilizar cuando el dispositivo se comunica conAWS IoTTrabajos, ejecute el `DescribeEndpoint` comando. Por ejemplo, si ejecuta este comando:

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

obtendrá un resultado similar al siguiente:

```
{  
    "endpointAddress": "a1b2c3d4e5f6g7-ats.iot.us-west-2.amazonaws.com"  
}
```

mediante el protocolo MQTT

Los dispositivos se pueden comunicar conAWS IoTTrabajos que utilizan el protocolo MQTT. Los dispositivos se suscriben a temas MQTT para recibir notificaciones de trabajos nuevos y respuestas del servicio Jobs de AWS IoT. Los dispositivos publican en temas MQTT para consultar o actualizar el estado de una ejecución de trabajo. Cada dispositivo tiene su propio tema MQTT general. Para obtener más información acerca de cómo publicar en temas de MQTT y suscribirse a ellos, consulte [Protocolos de comunicación de dispositivos \(p. 81\)](#).

Con este método de comunicación, el dispositivo utiliza su propio certificado y su propia clave privada para autenticarse enAWS IoTTrabajos.

Los dispositivos pueden suscribirse a los siguientes temas.`thing-name`Es el nombre del objeto asociado con el dispositivo.

- **\$aws/things/`thing-name`/jobs/notify**

Suscríbase a este tema para notificarle cuando se añade o se quita una ejecución de trabajo de la lista de ejecuciones de trabajo pendientes.

- **\$aws/things/`thing-name`/jobs/notify-next**

Suscríbase a este tema para notificarle cuándo cambia la siguiente ejecución de trabajo pendiente.

- **\$aws/things/`thing-name`/request-name/accepted**

El servicio Jobs de AWS IoT publica los mensajes de éxito y error en un tema de MQTT. El tema se forma añadiendo `accepted` o `rejected` al tema utilizado para realizar la solicitud. Aquí, `request-name`es el nombre de una solicitud, como `Get`, y el tema puede ser: `$aws/things/myThing/jobs/`

get.AWS IoTA continuación, Jobs publica mensajes de éxito en el\$aws/things/myThing/jobs/get/acceptedtema.

- **\$aws/things/*thing-name*/request-name/rejected**

Aquí,request-namees el nombre de una solicitud, comoGet. Si la solicitud ha fallado,AWS IoTJobs publica mensajes de error en el\$aws/things/myThing/jobs/get/rejectedtema.

También puede utilizar las siguientes API HTTPS:

- Actualizar el estado de una ejecución de trabajo llamando a la API [UpdateJobExecution](#).
- Consultar el estado de una ejecución de trabajo llamando a la API [DescribeJobExecution](#).
- Recuperar una lista de ejecuciones de trabajo pendientes llamando a la API [GetPendingJobExecutions](#).
- Recuperar la siguiente ejecución de trabajo pendiente llamando al[DescribeJobExecutionAPI](#) conjobIdcomo\$next.
- Obtener e iniciar la ejecución de trabajo pendiente siguiente llamando a la API [StartNextPendingJobExecution](#).

mediante HTTP Signature Version 4

Los dispositivos se pueden comunicar conAWS IoTTrabajos utilizando HTTP Signature Version 4 en el puerto 443. Este es el método utilizado por elAWSSDK y CLI. Para obtener más información acerca de estas herramientas, consulte[AWS CLIReferencia de comandos: iot-jobs-dataoAWSSDK y herramientasy](#) consulte el IoTJobsDataPlane para el idioma que prefiera.

Con este método de comunicación, el dispositivo utiliza credenciales de IAM para autenticarse conAWS IoTTrabajos.

Los siguientes comandos están disponibles a través de este 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 ...
```

mediante HTTP TLS

Los dispositivos se pueden comunicar conAWS IoTTrabajos utilizando HTTP TLS en el puerto 8443 utilizando un cliente de software de terceros que admite este protocolo.

Con este método, el dispositivo utiliza la autenticación basada en certificados X.509 (por ejemplo, utilizando su propio certificado y su propia clave privada).

Los siguientes comandos están disponibles a través de este método:

- [DescribeJobExecution](#)

- GetPendingJobExecutions
- StartNextPendingJobExecution
- UpdateJobExecution

Programación de dispositivos para trabajar con trabajos

En los ejemplos de esta sección se utiliza MQTT para ilustrar cómo funciona un dispositivo con el servicio Jobs de AWS IoT. También puede utilizar los comandos de la CLI o la API correspondientes. Para estos ejemplos, supondremos que un dispositivo llamado *MyThing* que se suscribe a los siguientes temas de MQTT:

- `$aws/things/MyThing/jobs/notify` (o `$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`

Si utiliza firma con código para AWS IoT, el código del dispositivo debe verificar la firma del archivo de código. La firma se encuentra en el documento de trabajo, en la propiedad `codeSign`. Para obtener más información sobre cómo verificar una firma en un archivo de código, consulte el [ejemplo de agente de dispositivo](#).

Temas

- [Flujo de trabajo del dispositivo \(p. 720\)](#)
- [Flujo de trabajo \(p. 721\)](#)
- [Notificaciones de trabajos \(p. 724\)](#)

Flujo de trabajo del dispositivo

Un dispositivo puede gestionar los trabajos que ejecuta mediante cualquiera de las siguientes formas.

- Obtención del siguiente trabajo
 1. Cuando un dispositivo está online, debe suscribirse al tema `notify-next` del dispositivo.
 2. Llame a la API de MQTT de [DescribeJobExecution \(p. 748\)](#) con `jobId $next` para obtener el siguiente trabajo, su documento de trabajo y otros detalles, incluido el estado guardado en `statusDetails`. Si el documento de trabajo tiene una forma en archivo de código, debe verificar la firma antes de seguir procesando la solicitud del trabajo.
 3. Llame a la API de MQTT de [UpdateJobExecution \(p. 749\)](#) para actualizar el estado del trabajo. También puede combinar este paso y el anterior en una llamada. El dispositivo puede llamar a [StartNextPendingJobExecution \(p. 747\)](#).
 4. Si lo prefiere, puede añadir un temporizador de pasos estableciendo un valor para `stepTimeoutInMinutes` si llama a [UpdateJobExecution \(p. 749\)](#) o [StartNextPendingJobExecution \(p. 747\)](#).
 5. Realice las acciones especificadas por el documento de trabajo con la API de MQTT de [UpdateJobExecution \(p. 749\)](#) para informar del progreso del trabajo.
 6. Siga monitorizando la ejecución del trabajo llamando a la API de MQTT [DescribeJobExecution \(p. 748\)](#) con este `jobId`. Si se elimina la ejecución del trabajo, [DescribeJobExecution \(p. 748\)](#) devuelve un objeto `ResourceNotFoundException`.

El dispositivo debería ser capaz de recuperarse a un estado válido si la ejecución de trabajo se cancela o se elimina cuando el dispositivo está ejecutando el trabajo.

7. Llame a la API de MQTT de [UpdateJobExecution \(p. 749\)](#) cuando termine con el trabajo para actualizar el estado del trabajo e informar del éxito o error.
8. El siguiente trabajo disponible para su ejecución (si lo hubiera) cambiará, puesto que el estado de la ejecución del trabajo ha cambiado a estado final. Se notifica al dispositivo que la siguiente ejecución de trabajo pendiente ha cambiado. En este momento, el dispositivo debe continuar como se describe en el paso 2.

Si el dispositivo sigue online, seguirá recibiendo notificaciones de la siguiente ejecución de trabajo pendiente, incluidos sus datos de ejecución del trabajo, cuando complete un trabajo o cuando se añada una nueva ejecución de trabajo pendiente. Cuando se produzca esto, el dispositivo seguirá como se describe en el paso 2.

- Selección a partir de trabajos disponibles

1. Cuando un dispositivo está online, debe suscribirse al tema `notify` del objeto.
2. Llame a la API de MQTT de [GetPendingJobExecutions \(p. 746\)](#) para obtener una lista de ejecuciones de trabajo pendientes.
3. Si la lista contiene una o más ejecuciones de trabajo, elija una.
4. Llame a la API de MQTT [DescribeJobExecution \(p. 748\)](#) para obtener el documento de trabajo y otros detalles, incluido cualquier estado guardado en `statusDetails`.
5. Llame a la API de MQTT de [UpdateJobExecution \(p. 749\)](#) para actualizar el estado del trabajo. Si el campo `includeJobDocument` está establecido en `true` en este comando, el dispositivo puede omitir el paso anterior y recuperar el documento de trabajo en este momento.
6. Si lo prefiere, puede añadir un temporizador de pasos estableciendo un valor para `stepTimeoutInMinutes` si llama a [UpdateJobExecution \(p. 749\)](#).
7. Realice las acciones especificadas por el documento de trabajo con la API de MQTT de [UpdateJobExecution \(p. 749\)](#) para informar del progreso del trabajo.
8. Siga monitorizando la ejecución del trabajo llamando a la API de MQTT [DescribeJobExecution \(p. 748\)](#) con este `jobId`. Si la ejecución de trabajo se cancela o se elimina al mismo tiempo que el dispositivo está ejecutando el trabajo, el dispositivo debería ser capaz de recuperarse a un estado válido.
9. Llame a la API de MQTT de [UpdateJobExecution \(p. 749\)](#) cuando termine con el trabajo para actualizar el estado del trabajo e informar del éxito o error.

Si el dispositivo continúa online, se le notificarán todas las ejecuciones de trabajo pendientes cuando una nueva ejecución de trabajo pendiente esté disponible. Cuando se produzca esto, el dispositivo podrá seguir como se describe en el paso 2.

Si el dispositivo no puede ejecutar el trabajo, debe llamar a la API de MQTT de [UpdateJobExecution \(p. 749\)](#) para actualizar el estado del trabajo a `REJECTED`.

Flujo de trabajo

A continuación se muestran los distintos pasos del flujo de trabajo de trabajos, desde iniciar un nuevo trabajo hasta informar del estado de finalización de la ejecución de un trabajo.

Inicio de un nuevo trabajo

Cuando se crea un nuevo trabajo, AWS IoTJobs publica un mensaje en el \$aws/things/`thing-name`/jobs/notify tema para cada dispositivo de destino.

El mensaje contiene la siguiente información:

```
{  
    "timestamp":1476214217017,  
    "jobs":{  
        "QUEUED": [  
            {  
                "jobId":"0001",  
                "queuedAt":1476214216981,  
                "lastUpdatedAt":1476214216981,  
                "versionNumber" : 1  
            }  
        ]  
    }  
}
```

El dispositivo recibe este mensaje en el tema '\$aws/things/*thingName*/jobs/notify' cuando la ejecución del trabajo está en la cola.

obtener información de trabajo

Para obtener más información sobre la ejecución de un trabajo, el dispositivo llama a la API de MQTT [DescribeJobExecution \(p. 748\)](#) con el campo `includeJobDocument` establecido en `true` (el valor predeterminado).

Si la solicitud es correcta, el servicio Jobs de AWS IoT publica un mensaje en el tema \$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 >  
        }  
    }  
}
```

Si se produce un error en la solicitud, el servicio Jobs de AWS IoT publica un mensaje en el tema \$aws/things/MyThing/jobs/0023/get/rejected.

El dispositivo ahora tiene el documento de trabajo que puede usar para realizar las operaciones remotas para el trabajo. Si el documento de trabajo contiene una URL prefirmada de Amazon S3, el dispositivo puede usar esa URL para descargar los archivos necesarios para el trabajo.

Informe del estado de la ejecución de trabajo

A medida que el dispositivo ejecute el trabajo, puede llamar a la API de MQTT de [UpdateJobExecution \(p. 749\)](#) para actualizar el estado de la ejecución de trabajo.

Por ejemplo, un dispositivo puede actualizar el estado de ejecución de trabajo a IN_PROGRESS mediante la publicación del siguiente mensaje en el tema \$aws/things/MyThing/jobs/0023/update:

```
{  
    "status":"IN_PROGRESS",
```

```
"statusDetails": {  
    "progress": "50%"  
},  
"expectedVersion": "1",  
"clientToken": "client001"  
}
```

Jobs responde mediante la publicación de un mensaje en el tema `$aws/things/MyThing/jobs/0023/update/accepted` o `$aws/things/MyThing/jobs/0023/update/rejected`:

```
{  
    "clientToken": "client001",  
    "timestamp": 1476289222841  
}
```

El dispositivo puede combinar las dos solicitudes anteriores llamando a [StartNextPendingJobExecution \(p. 747\)](#). Esto obtiene e inicia la ejecución del trabajo pendiente y permite al dispositivo actualizar el estado de ejecución del trabajo. Esta solicitud también devuelve el documento de trabajo cuando hay una ejecución de trabajo pendiente.

Si el trabajo contiene un valor [TimeoutConfig](#), el temporizador en curso empezará a funcionar. También puede establecer un temporizador de pasos para la ejecución de un trabajo especificando un valor para `stepTimeoutInMinutes` al llamar a [UpdateJobExecution](#). El temporizador de pasos se aplica únicamente a la ejecución del trabajo que actualice. Puede establecer un nuevo valor para este temporizador cada vez que actualice la ejecución de un trabajo. También puede crear un temporizador de pasos al llamar a [StartNextPendingJobExecution](#). Si la ejecución del trabajo permanece en estado `IN_PROGRESS` durante un periodo superior a este intervalo del temporizador de pasos, generará un error y cambiará al estado terminal `TIMED_OUT`. El temporizador de pasos no tiene ningún efecto en el temporizador en curso que establezca al crear un trabajo.

El campo `status` se puede definir en `IN_PROGRESS`, `SUCCEEDED` o `FAILED`. No puede actualizar el estado de una ejecución de trabajo que ya está en estado terminal.

ejecución de informe completada

Cuando el dispositivo ha acabado de ejecutar el trabajo, llama a la API de MQTT de [UpdateJobExecution \(p. 749\)](#). Si el trabajo se realizó correctamente, establezca `status` en `SUCCEEDED` y, en la carga del mensaje, en `statusDetails`, añada otra información acerca del trabajo, como pares nombre-valor. Los temporizadores en curso y de pasos finalizan cuando se completa la ejecución del trabajo.

Por ejemplo:

```
{  
    "status": "SUCCEEDED",  
    "statusDetails": {  
        "progress": "100%"  
    },  
    "expectedVersion": "2",  
    "clientToken": "client-001"  
}
```

Si el trabajo no se realizó correctamente, establezca `status` en `FAILED` y, en `statusDetails`, añada información acerca del error que se ha producido:

```
{  
    "status": "FAILED",  
    "statusDetails": {  
        "error": "Job execution failed due to system error."  
    }  
}
```

```
    "errorCode":"101",
    "errorMsg":"Unable to install update"
},
"expectedVersion":"2",
"clientToken":"client-001"
}
```

Note

El atributo `statusDetails` puede contener cualquier número de pares nombre-valor.

Cuando el servicio Jobs de AWS IoT recibe esta actualización, publica un mensaje en el tema `$aws/things/MyThing/jobs/notify` para indicar que la ejecución del trabajo se ha completado:

```
{
  "timestamp":1476290692776,
  "jobs":{}
}
```

Trabajos adicionales

Si hay otras ejecuciones de trabajo pendientes para el dispositivo, se incluyen en el mensaje publicado en `$aws/things/MyThing/jobs/notify`.

Por ejemplo:

```
{
  "timestamp":1476290692776,
  "jobs":{
    "QUEUED":[{
      "jobId":"0002",
      "queuedAt":1476290646230,
      "lastUpdatedAt":1476290646230
    }],
    "IN_PROGRESS":[{
      "jobId":"0003",
      "queuedAt":1476290646230,
      "lastUpdatedAt":1476290646230
    }]
  }
}
```

Notificaciones de trabajos

El servicio Jobs de AWS IoT publica mensajes de MQTT para temas reservados cuando los trabajos están pendientes o cuando cambia la primera ejecución del trabajo en la lista. Los dispositivos pueden realizar un seguimiento de los trabajos pendientes suscribiéndose a estos temas.

Tipos de notificación de Job

Las notificaciones de trabajo se publican en temas MQTT como cargas JSON. Existen dos tipos de notificaciones:

- Una `ListNotification` contiene una lista de no más de 10 ejecuciones de trabajos pendientes. Las ejecuciones de trabajo de esta lista tienen valores de estado `IN_PROGRESS` o `QUEUED`. Se almacenan por estado (ejecuciones de trabajo `IN_PROGRESS` antes que las ejecuciones de trabajo `QUEUED`) y, a continuación, por las veces que se incluyeron en la cola.

Una `ListNotification` se publica siempre que se cumple uno de los siguientes criterios.

- Se pone en cola una nueva ejecución de trabajo o se cambia a un estado no terminal (`IN_PROGRESS` o `QUEUED`).
- Una ejecución de trabajo antigua cambia a un estado terminal (`FAILED`, `SUCCEEDED`, `CANCELED`, `TIMED_OUT`, `REJECTED` o `REMOVED`).
- `NextNotification` contiene información resumida sobre la ejecución del trabajo a continuación en la cola.

Se publica una `NextNotification` siempre que cambia la ejecución del primer trabajo de la lista.

- Se añade a la lista una nueva ejecución del trabajo como `QUEUED` y se coloca el primero en la lista.
- El estado de la ejecución de un trabajo existente que no estaba en el primer lugar en la lista cambia de `QUEUED` a `IN_PROGRESS` y pasa a colocarse en primer lugar en la lista. (Esto sucede cuando no hay otras ejecuciones de trabajos de `IN_PROGRESS` en la lista o cuando la ejecución del trabajo cuyo estado cambia de `QUEUED` a `IN_PROGRESS` estaba en la cola antes que cualquier otra ejecución de trabajo de `IN_PROGRESS` de la lista.)
- El estado de la ejecución del trabajo existente que se encuentra en el primer lugar de la lista cambia a un estado terminal y se quita de la lista.

Para obtener más información acerca de cómo publicar en temas de MQTT y suscribirse a ellos, consulte [the section called “Protocolos de comunicación de dispositivos” \(p. 81\)](#).

Note

Las notificaciones no están disponibles cuando se usa HTTP Signature Version 4 o HTTP TLS para comunicarse con trabajos.

Job pendiente

El servicio Jobs de AWS IoT publica un mensaje en un tema de MQTT cuando se añade un trabajo a la lista de ejecuciones de trabajos pendientes para un objeto o se quita un trabajo de dicha lista, o cuando cambia la primera ejecución del trabajo en la lista:

- `$aws/things/thingName/jobs/notify`
- `$aws/things/thingName/jobs/notify-next`

Los mensajes contienen las siguientes cargas de ejemplo:

`$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"}
  }
}
```

Los valores del estado de ejecución de trabajo posibles son QUEUED, IN_PROGRESS, FAILED, SUCCEEDED, CANCELED, TIMED_OUT, REJECTED y REMOVED.

La siguiente serie de ejemplos muestra las notificaciones que se publican en cada tema a medida que se crean las ejecuciones de trabajo y se cambian de un estado a otro.

En primer lugar se crea un trabajo llamado job1. La notificación se publica en el tema jobs/notify:

```
{
  "timestamp": 1517016948,
  "jobs": {
    "QUEUED": [
      {
        "jobId": "job1",
        "queuedAt": 1517016947,
        "lastUpdatedAt": 1517016947,
        "executionNumber": 1,
        "versionNumber": 1
      }
    ]
  }
}
```

La notificación se publica en el tema jobs/notify-next:

```
{
  "timestamp": 1517016948,
  "execution": {
    "jobId": "job1",
    "status": "QUEUED",
    "queuedAt": 1517016947,
    "lastUpdatedAt": 1517016947,
    "versionNumber": 1,
    "executionNumber": 1,
    "jobDocument": {
      "operation": "test"
    }
  }
}
```

Cuando se crea otro trabajo (job2), esta notificación se publica en el tema 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  
        }  
    ]  
}
```

No se publica una notificación en el tema `jobs/notify-next` porque el siguiente trabajo de la cola (`job1`) no ha cambiado. Cuando `job1` comienza a ejecutarse, su estado cambia a `IN_PROGRESS`. No se publican notificaciones ya que la lista de trabajos y el siguiente trabajo en la cola no han cambiado.

Cuando se añade un tercer trabajo (`job3`), esta notificación se publica en el tema `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  
            }  
        ]  
    }  
}
```

No se publica una notificación en el tema `jobs/notify-next` porque el siguiente trabajo en la cola sigue siendo `job1`.

Cuando `job1` finaliza, su estado cambia a `SUCCEEDED` y se publica esta notificación en el tema `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  
      }  
    ]  
  }  
}
```

En este punto, job1 se ha eliminado de la cola y el siguiente trabajo que ejecutar es job2. La notificación se publica en el tema `jobs/notify-next`:

```
{  
  "timestamp": 1517186269,  
  "execution": {  
    "jobId": "job2",  
    "status": "QUEUED",  
    "queuedAt": 1517017191,  
    "lastUpdatedAt": 1517017191,  
    "versionNumber": 1,  
    "executionNumber": 1,  
    "jobDocument": {  
      "operation": "test"  
    }  
  }  
}
```

Si job3 tiene que empezar a ejecutarse antes que job2 (lo cual no se recomienda), el estado de job3 puede cambiarse a `IN_PROGRESS`. Si esto sucede, job2 deja de ser el siguiente en la cola y se publica esta notificación en el tema `jobs/notify-next`:

```
{  
  "timestamp": 1517186779,  
  "execution": {  
    "jobId": "job3",  
    "status": "IN_PROGRESS",  
    "queuedAt": 1517017905,  
    "startedAt": 1517186779,  
    "lastUpdatedAt": 1517186779,  
    "versionNumber": 2,  
    "executionNumber": 1,  
    "jobDocument": {  
      "operation": "test"  
    }  
  }  
}
```

No se publica ninguna notificación en el tema `jobs/notify`, dado que no se ha añadido o eliminado ningún trabajo.

Si el dispositivo rechaza job2 y actualiza su estado a REJECTED, se publica esta notificación en el tema jobs/notify:

```
{  
  "timestamp": 1517189392,  
  "jobs": [  
    {"IN_PROGRESS": [  
      {"jobId": "job3",  
       "queuedAt": 1517017905,  
       "lastUpdatedAt": 1517186779,  
       "startedAt": 1517186779,  
       "executionNumber": 1,  
       "versionNumber": 2  
     }  
    ]  
  }  
}
```

Si job3 (que aún está en curso) se elimina de forma forzada, esta notificación se publica en el tema jobs/notify:

```
{  
  "timestamp": 1517189551,  
  "jobs": {}  
}
```

En este momento, la cola está vacía. La notificación se publica en el tema jobs/notify-next:

```
{  
  "timestamp": 1517189551  
}
```

AWS IoT API de trabajos

AWS IoT API de trabajos se puede utilizar para cualquiera de las siguientes categorías:

- Las operaciones de la API se utilizan para tareas administrativas como la administración y el control de trabajos. Este es el plano de control.
- Operaciones de API utilizadas para dispositivos que ejecutan esos trabajos. Este es el plano de datos, que permite enviar y recibir datos.

La administración y el control de Job utilizan una API de protocolo HTTPS. Los dispositivos pueden usar una API de protocolo HTTPS o MQTT. La API de plano de control está diseñada para un volumen reducido de llamadas normalmente cuando se crean trabajos y se hace un seguimiento de ellos. Normalmente, abre una conexión para una solicitud única y, a continuación, cierra la conexión después de que se reciba la respuesta. El plano de datos HTTPS y la API MQTT permiten realizar sondeos prolongados. Estas operaciones de la API están diseñadas para cantidades grandes de tráfico que pueden escalar millones de dispositivos.

Cada API HTTPS de Jobs de AWS IoT tiene un comando correspondiente que le permite llamar a la API desde la AWS CLI. Los comandos están en minúsculas, con guiones entre las palabras que conforman el nombre de la API. Por ejemplo, puede invocar la API CreateJob en la CLI escribiendo lo siguiente:

```
aws iot create-job ...
```

Si se produce un error durante una operación, obtiene una respuesta de error que contiene información acerca del error.

ErrorResponse

Contiene información acerca de un error que se produjo durante una operación del servicio Jobs de AWS IoT.

En el siguiente ejemplo se muestra la sintaxis de esta operación:

```
{  
    "code": "ErrorCode",  
    "message": "string",  
    "clientToken": "string",  
    "timestamp": timestamp,  
    "executionState": JobExecutionState  
}
```

A continuación se muestra una descripción de este `ErrorResponse`:

code

`ErrorCode` se puede establecer en:

InvalidTopic

La solicitud se envió a un tema en el espacio de nombres de Jobs de AWS IoT que no está asignado a ninguna API.

InvalidJson

El contenido de la solicitud podría no interpretarse como un formato JSON con codificación UTF-8 válida.

InvalidRequest

El contenido de la solicitud no es válido. Por ejemplo, se devuelve este código cuando una solicitud `UpdateJobExecution` contiene detalles de estado no válido. El mensaje contiene detalles acerca del error.

InvalidStateTransition

Una actualización intentó cambiar la ejecución de trabajo a un estado que no es válido debido al estado actual de la ejecución de trabajo (por ejemplo, un intento de cambiar una solicitud con un estado `SUCCEEDED` a un estado `IN_PROGRESS`). En este caso, el cuerpo del mensaje de error también contiene el campo `executionState`.

ResourceNotFound

El `JobExecution` especificado por el tema de la solicitud no existe.

VersionMismatch

La versión esperada especificada en la solicitud no coincide con la versión de la ejecución del trabajo en el servicio Jobs de AWS IoT. En este caso, el cuerpo del mensaje de error también contiene el campo `executionState`.

InternalError

Se ha producido un error interno al procesar la solicitud.

RequestThrottled

La solicitud se ha limitado.

TerminalStateReached

Se produce cuando un comando para describir un trabajo se realiza en un trabajo que está en un estado terminal.

message

Una cadena de mensajes de error.

clientToken

Una cadena arbitraria utilizada para correlacionar una solicitud con su respuesta.

timestamp

El tiempo, en segundos, desde la fecha de inicio.

executionState

Un objeto `JobExecutionState`. Este campo se incluye solo cuando el campo `code` tiene el valor `InvalidStateTransition` o `VersionMismatch`. Esto hace que no sea necesario en esos casos realizar una solicitud `DescribeJobExecution` independiente para obtener los datos de estado de ejecución de trabajo actuales.

A continuación se enumeran las API de trabajos y los tipos de datos.

- [API de administración y control de trabajo y tipos de datos \(p. 731\)](#)
- [API de MQTT y HTTPS de dispositivo de trabajo y tipos de datos \(p. 745\)](#)

API de administración y control de trabajo y tipos de datos

Los siguientes comandos están disponibles para la administración de Job y el control en la CLI a través del protocolo HTTPS.

- [Tipos de datos de administración y control de trabajo \(p. 731\)](#)
- [Operaciones de API de administración y control de Job \(p. 734\)](#)

Para determinar el `endpoint-url` parámetro para los comandos de la CLI, ejecute este comando.

```
aws iot describe-endpoint --endpoint-type=iot:Jobs
```

Este comando devuelve la siguiente salida.

```
{  
  "endpointAddress": "account-specific-prefix.jobs.iot.aws-region.amazonaws.com"  
}
```

Note

El punto de enlace Jobs de no admite ALPNz-amzn-[http-ca](#).

Tipos de datos de administración y control de trabajo

Las aplicaciones de administración y control utilizan los siguientes tipos de datos para comunicarse con AWS IoT Trabajos.

Trabajo

El objeto `Job` contiene detalles acerca de un trabajo. A continuación se muestra la sintaxis:

```
{  
    "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 obtener más información, consulte [Jobo job](#)

JobSummary

El objeto `JobSummary` contiene un resumen de trabajos. A continuación se muestra la sintaxis:

```
{
```

```
"jobArn": "string",
"jobId": "string",
"status": "IN_PROGRESS|CANCELED|SUCCEEDED",
"targetSelection": "CONTINUOUS|SNAPSHOT",
"thingGroupId": "string",
"createdAt": timestamp,
"lastUpdatedAt": timestamp,
"completedAt": timestamp
}
```

Para obtener más información, consulte [JobSummary](#) o [job-summary](#).

JobExecution

El objeto `JobExecution` representa la ejecución de un trabajo en un dispositivo. A continuación se muestra la sintaxis:

```
{
    "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 obtener más información, consulte [JobExecution](#) o [job-execution](#).

JobExecutionSummary

La `JobExecutionSummary` contiene información de resumen de ejecución de trabajos. A continuación se muestra la sintaxis:

```
{
    "executionNumber": 1234567890,
    "queuedAt": timestamp,
    "lastUpdatedAt": timestamp,
    "startedAt": timestamp,
    "status": "QUEUED|IN_PROGRESS|FAILED|SUCCEEDED|CANCELED|TIMED_OUT|REJECTED|REMOVED"
}
```

Para obtener más información, consulte [JobExecutionSummary](#) o [job-execution-summary](#).

JobExecutionSummaryForJob

El objeto `JobExecutionSummaryForJob` contiene un resumen de información acerca de las ejecuciones de trabajo para un trabajo específico. A continuación se muestra la sintaxis:

```
{
```

```
"executionSummaries": [
    {
        "thingArn": "arn:aws:iot:us-west-2:123456789012:thing/MyThing",
        "jobExecutionSummary": {
            "status": "IN_PROGRESS",
            "lastUpdatedAt": 1549395301.389,
            "queuedAt": 1541526002.609,
            "executionNumber": 1
        }
    },
    ...
]
```

Para obtener más información, consulte [JobExecutionSummaryForJob](#) o [job-execution-summary-for-job](#).

JobExecutionSummaryForThing

El objeto `JobExecutionSummaryForThing` contiene un resumen de información acerca de una ejecución de trabajo en un objeto específico. A continuación se muestra la sintaxis:

```
{
    "executionSummaries": [
        {
            "jobExecutionSummary": {
                "status": "IN_PROGRESS",
                "lastUpdatedAt": 1549395301.389,
                "queuedAt": 1541526002.609,
                "executionNumber": 1
            },
            "jobId": "MyThingJob"
        },
        ...
    ]
}
```

Para obtener más información, consulte [JobExecutionSummaryForThing](#) o [job-execution-summary-for-thing](#).

Operaciones de API de administración y control de Job

Utilice las siguientes operaciones de API o comandos de CLI:

AssociateTargetsWithJob

Asocia un grupo a un trabajo continuo. Deben cumplirse los siguientes criterios:

- El trabajo debe haberse creado con el campo `targetSelection` establecido en `CONTINUOUS`.
- El estado del trabajo debe ser actualmente `IN_PROGRESS`.
- El número total de destinos asociados con un trabajo no debe ser superior a 100.

HTTPS request

```
POST /jobs/jobId/targets
{
    "targets": [ "string" ],
```

```
    "comment": "string"
}
```

Para obtener más información, 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 formato:

```
{
  "targets": [
    "string"
  ],
  "jobId": "string",
  "comment": "string"
}
```

Para obtener más información, consulte [associate-targets-with-job](#).

CancelJob

Cancela un trabajo.

HTTPS request

```
PUT /jobs/jobId/cancel

{
  "force": boolean,
  "comment": "string",
  "reasonCode": "string"
}
```

Para obtener más información, 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 formato:

```
{
  "jobId": "string",
  "force": boolean,
  "comment": "string"
```

}

Para obtener más información, consulte [cancel-job](#).

CancelJobExecution

Cancela la ejecución de un trabajo en un dispositivo.

HTTPS request

```
PUT /things/thingName/jobs/jobId/cancel

{
  "force": boolean,
  "expectedVersion": "string",
  "statusDetails": {
    "string": "string"
    ...
  }
}
```

Para obtener más información, 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 formato:

```
{
  "jobId": "string",
  "thingName": "string",
  "force": boolean,
  "expectedVersion": long,
  "statusDetails": {
    "string": "string"
  }
}
```

Para obtener más información, consulte [cancel-job-execution](#).

CreateJob

Crea un trabajo. Puede proporcionar el documento de trabajo como enlace a un archivo en un bucket de Amazon S3 (`documentSource` parámetro) o en el cuerpo de la solicitud (`document` parámetro).

Un trabajo puede ser continuo si se establece el parámetro opcional `targetSelection` en `CONTINUOUS`. (El valor predeterminado es `SNAPSHOT`). Un trabajo continuo puede utilizarse para integrar o actualizar dispositivos a medida que se añaden a un grupo, ya que continúa ejecutándose, y lo hace en objetos añadidos recientemente, incluso después de que los objetos del grupo que en el momento en el que se creó el trabajo hubieran completado el trabajo.

Un trabajo puede tener un valor [TimeoutConfig](#) opcional que establece el valor del temporizador en curso. El temporizador en curso no se puede actualizar y se aplica a todas las ejecuciones del trabajo.

Se realizan las siguientes validaciones en los argumentos para la API [CreateJob](#):

- El argumento `targets` debe ser una lista de ARN de grupo de objetos u objetos válidos. Todos los objetos y grupos de cosas deben estar en tuCuenta de AWS.
- La `documentSource` debe ser una URL de Amazon S3 válida para un documento de trabajo. Las URL de Amazon S3 tienen la siguiente forma:`https://s3.amazonaws.com/bucketName/objectName`.
- El documento almacenado en la URL especificada por el argumento `documentSource` debe ser un documento JSON con codificación UTF-8.
- El tamaño del documento de trabajo está limitado a 32 KB debido al límite del tamaño de un mensaje MQTT (128 KB) y el cifrado.
- La `jobId` debe ser único en suCuenta de 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 obtener más información, 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 formato:

```
{
    "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 obtener más información, consulte [create-job](#).

DeleteJob

Elimina un trabajo y sus ejecuciones de trabajo relacionadas.

La eliminación de un trabajo puede tardar tiempo, en función del número de ejecuciones de trabajo creadas para el trabajo y otros factores diversos. Aunque el trabajo se está eliminando, el estado del trabajo se muestra como "DELETION_IN_PROGRESS". Si se intenta eliminar o cancelar un trabajo cuyo estado ya es DELETION_IN_PROGRESS, se producirá un error.

HTTPS request

```
DELETE /jobs/jobId?force=force
```

Para obtener más información, consulte [DeleteJob](#).

CLI syntax

```
aws iot delete-job \  
--job-id <value> \  
[--force | --no-force] \  
[--cli-input-json <value>] \  
[--generate-cli-skeleton]
```

cli-input-json formato:

```
{  
"jobId": "string",  
"force": boolean  
}
```

Para obtener más información, consulte [delete-job](#).

DeleteJobExecution

Elimina una ejecución de trabajo.

HTTPS request

```
DELETE /things/thingName/jobs/jobId/executionNumber/executionNumber?force=force
```

Para obtener más información, 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 formato:

```
{  
"jobId": "string",  
"thingName": "string",  
"executionNumber": long,  
"force": boolean  
}
```

Para obtener más información, consulte [delete-job-execution](#).

DescribeJob

Obtiene los detalles del trabajo especificado.

HTTPS request

```
DELETE /jobs/jobId?force=force
```

Para obtener más información, consulte [DeleteJob](#).

CLI syntax

```
aws iot delete-job \  
--job-id <value> \  
[--force | --no-force] \  
[--cli-input-json <value>] \  
[--generate-cli-skeleton]
```

cli-input-json formato:

```
{  
"jobId": "string",  
"force": boolean  
}
```

Para obtener más información, consulte [delete-job](#).

DescribeJob

Obtiene los detalles de la ejecución del trabajo.

HTTPS request

```
GET /jobs/jobId
```

Para obtener más información, consulte [DescribeJob](#).

CLI syntax

```
aws iot describe-job \  
--job-id <value> \  
[--cli-input-json <value>] \  
[--generate-cli-skeleton]
```

cli-input-json formato:

```
{  
"jobId": "string"  
}
```

Para obtener más información, consulte [describe-job](#).

DescribeJobExecution

Obtiene los detalles de una ejecución de trabajo. El estado de la ejecución del trabajo debe ser SUCCEEDED o FAILED.

HTTPS request

```
GET /things/thingName/jobs/jobId?executionNumber=executionNumber
```

Para obtener más información, 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 formato:

```
{  
"jobId": "string",  
"thingName": "string",  
"executionNumber": long  
}
```

Para obtener más información, consulte [describe-job-execution](#).

GetJobDocument

Obtiene el documento de trabajo para un trabajo.

Note

Las URL de marcador de posición no se reemplazan por las URL de Amazon S3 prefijadas en el documento devuelto. Las URL prefijadas se generan solo cuando el servicio Jobs de AWS IoT recibe una solicitud a través de MQTT.

HTTPS request

```
GET /jobs/jobId/job-document
```

Para obtener más información, consulte [GetJobDocument](#).

CLI syntax

```
aws iot get-job-document \  
--job-id <value> \  
[--cli-input-json <value>] \  
[--generate-cli-skeleton]
```

cli-input-json formato:

```
{  
"jobId": "string"  
}
```

Para obtener más información, consulte [get-job-document](#).

ListJobExecutionsForJob

Obtiene una lista de ejecuciones de trabajo para un trabajo.

HTTPS request

```
GET /jobs/jobId/things?status=status&maxResults=maxResults&nextToken=nextToken
```

Para obtener más información, 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 formato:

```
{  
"jobId": "string",  
"status": "string",  
"maxResults": "integer",  
"nextToken": "string"  
}
```

Para obtener más información, consulte [list-job-executions-for-job](#).

[ListJobExecutionsForThing](#)

Obtiene una lista de ejecuciones de trabajo para un objeto.

HTTPS request

```
GET /things/thingName/jobs?status=status&maxResults=maxResults&nextToken=nextToken
```

Para obtener más información, 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 formato:

```
{  
"thingName": "string",  
"status": "string",  
"maxResults": "integer",  
"nextToken": "string"  
}
```

Para obtener más información, consulte [list-job-executions-for-thing](#).

ListJobs

Obtiene una lista de trabajos en elCuenta de AWS.

HTTPS request

```
GET /jobs?  
status=status&targetSelection=targetSelection&thingGroupName=thingGroupName&thingGroupId=thingGroupId
```

Para obtener más información, 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 formato:

```
{  
    "status": "string",  
    "targetSelection": "string",  
    "maxResults": "integer",  
    "nextToken": "string",  
    "thingGroupName": "string",  
    "thingGroupId": "string"  
}
```

Para obtener más información, consulte [list-jobs](#).

UpdateJob

Actualiza los campos admitidos del trabajo especificado. Los valores actualizados de timeoutConfig surten efecto solo en ejecuciones recientemente en curso. Las ejecuciones en curso en ese momento siguen haciéndolo con la antigua configuración de tiempo de espera.

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 obtener más información, 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 formato:

```
{
    "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 obtener más información, consulte [update-job](#).

API de MQTT y HTTPS de dispositivo de trabajo y tipos de datos

Los siguientes comandos están disponibles a través de los protocolos MQTT y HTTPS. Utilice estas operaciones de API en el plano de datos para los dispositivos que ejecutan los trabajos.

Tipos de datos de MQTT y HTTPS de dispositivo de trabajo

Los siguientes tipos de datos se utilizan para comunicarse con el servicio de Jobs de AWS IoT a través de protocolos HTTPS y MQTT.

JobExecution

Contiene datos acerca de la ejecución de trabajo. En el siguiente ejemplo se muestra la sintaxis:

```
{  
    "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 obtener más información, consulte [JobExecution](#) o [job-execution](#).

JobExecutionState

Contiene datos acerca del estado de una ejecución de trabajo. En el siguiente ejemplo se muestra la sintaxis:

```
{  
    "status": "QUEUED|IN_PROGRESS|FAILED|SUCCEEDED|CANCELED|TIMED_OUT|REJECTED|REMOVED",  
    "statusDetails": {  
        "string": "string"  
    }  
    ...  
    "versionNumber": "number"  
}
```

Para obtener más información, consulte [JobExecutionState](#) o [job-execution-state](#).

JobExecutionSummary

Contiene una subred de información acerca de una ejecución de trabajo. En el siguiente ejemplo se muestra la sintaxis:

```
{  
    "jobId": "string",
```

```
"queuedAt": timestamp,  
"startedAt": timestamp,  
"lastUpdatedAt": timestamp,  
"versionNumber": "number",  
"executionNumber": long  
}
```

Para obtener más información, consulte [JobExecutionSummary](#) o [job-execution-summary](#).

Obtenga más información sobre las operaciones de la API de MQTT y HTTPS en las siguientes secciones:

- [API de MQTT de dispositivo de trabajo \(p. 746\)](#)
- [API HTTP de dispositivo de trabajo \(p. 751\)](#)

API de MQTT de dispositivo de trabajo

Puede emitir comandos de dispositivos de trabajos publicando mensajes MQTT en el[Temas reservados utilizados para los comandos Trabajos \(p. 107\)](#).

El cliente del lado del dispositivo debe estar suscrito a los temas de mensajes de respuesta de estos comandos. Si utiliza elAWS IoTDevice Client, el dispositivo se suscribirá automáticamente a los temas de respuesta. Esto significa que el agente de mensajes publicará temas de mensajes de respuesta en el cliente que publicó el mensaje de comando, independientemente de que su cliente se haya suscrito a los temas del mensaje de respuesta o no. Estos mensajes de respuesta no pasan por el agente de mensajes y otros clientes o reglas no pueden suscribirlos.

Al suscribirse al trabajo y jobExecutiontemas de eventos para su solución de monitoreo de flotas, active primero[eventos de trabajo y ejecución de trabajo \(p. 1097\)](#)para recibir cualquier evento en el lado de la nube. Mensajes de progreso del Job que se procesan a través del agente de mensajes y que pueden ser utilizados porAWS IoTLas reglas se publican como[Eventos de trabajos \(p. 1108\)](#). Dado que el agente de mensajes publica mensajes de respuesta, incluso sin una suscripción explícita a ellos, el cliente debe estar configurado para recibir e identificar los mensajes que recibe. Su cliente también debe confirmar que elthingNameen el tema del mensaje entrante se aplica al nombre de la cosa del cliente antes de que el cliente actúe sobre el mensaje.

Note

Mensajes queAWS IoTenvía en respuesta a los mensajes de comando de la API de MQTT Jobs que se cargan a su cuenta, independientemente de que se haya suscrito a ellos explícitamente o no.

A continuación se muestran las operaciones de la API de MQTT y su sintaxis de solicitudes y respuestas. Todas las operaciones de API de MQTT tienen los parámetros siguientes:

clientToken

Un token de cliente opcional que puede utilizarse para correlacionar solicitudes y respuestas.
Introduzca un valor arbitrario aquí y se reflejará en la respuesta.

timestamp

El tiempo en segundos, desde la fecha de inicioépoca, cuando se envió el mensaje.

GetPendingJobExecutions

Obtiene la lista de todos los trabajos que no están en un estado terminal para un objeto específico.

Para invocar esta API, publique un mensaje en \$aws/things/*thingName*/jobs/get.

Carga de solicitud:

```
{ "clientToken": "string" }
```

El agente de mensajes publicará \$aws/things/*thingName*/jobs/get/accepted y \$aws/things/*thingName*/jobs/get/rejected incluso sin una suscripción específica a ellos. Sin embargo, para que su cliente reciba los mensajes, debe estar escuchándolos. Para obtener más información, consulte [la nota sobre los mensajes de la API de trabajos \(p. 746\)](#).

Carga de respuesta:

```
{
  "inProgressJobs" : [ JobExecutionSummary ... ],
  "queuedJobs" : [ JobExecutionSummary ... ],
  "timestamp" : 1489096425069,
  "clientToken" : "client-001"
}
```

Donde `inProgressJobs` y `queuedJobs` devuelven una lista de [JobExecutionSummary \(p. 745\)](#) objetos que tienen estado de `IN_PROGRESS` o `QUEUED`.

StartNextPendingJobExecution

Obtiene y comienza la siguiente ejecución de trabajo pendiente para un objeto (estado `IN_PROGRESS` o `QUEUED`).

- Ejecuciones de trabajos con estatus `IN_PROGRESS` se devuelven primero.
- Las ejecuciones de trabajo se devuelven en el orden en el que se crearon.
- Si la siguiente ejecución de trabajos pendientes es `QUEUED`, su estado cambia a `IN_PROGRESS` y los detalles de estado de la ejecución del trabajo se establecen según lo especificado.
- Si la siguiente ejecución de trabajos pendientes de inicio ya está `IN_PROGRESS`, sus detalles de estado no se modifican.
- Si no hay ejecuciones de trabajo pendientes, la respuesta no incluirá el campo `execution`.
- Si lo prefiere, puede crear un temporizador de pasos estableciendo un valor para la propiedad `stepTimeoutInMinutes`. Si no actualiza el valor de esta propiedad mediante la ejecución de `UpdateJobExecution`, la ejecución del trabajo agotará el tiempo de espera cuando venza el temporizador de pasos.

Para invocar esta API, publique un mensaje en \$aws/things/*thingName*/jobs/start-next.

Carga de solicitud:

```
{
  "statusDetails": {
    "string": "job-execution-state"
    ...
  },
  "stepTimeoutInMinutes": long,
  "clientToken": "string"
}
```

statusDetails

Un conjunto de pares nombre-valor que describen el estado de ejecución del trabajo. Si no se especifica, `statusDetails` no se modifica.

stepTimeOutInMinutes

Especifica la cantidad de tiempo que tiene este dispositivo para finalizar la ejecución de este trabajo. Si el estado de ejecución del trabajo no se ha establecido en un estado terminal antes

de que este temporizador venza o antes de que se restablezca el temporizador (llamando a `UpdateJobExecution`, estableciendo el estado en `IN_PROGRESS` y especificando un valor de tiempo de espera nuevo en el campo `stepTimeoutInMinutes`) el estado de ejecución se establece automáticamente en `TIMED_OUT`. Configurar este tiempo de espera no tiene ningún efecto en el tiempo de espera de la ejecución de ese trabajo que se pueda haber especificado cuando se creó el trabajo (`CreateJob` con el campo `timeoutConfig`).

El agente de mensajes publicará \$aws/things/*thingName*/jobs/start-next/accepted y \$aws/things/*thingName*/jobs/start-next/rejected incluso sin una suscripción específica a ellos. Sin embargo, para que su cliente reciba los mensajes, debe estar escuchándolos. Para obtener más información, consulte [la nota sobre los mensajes de la API de trabajos \(p. 746\)](#).

Carga de respuesta:

```
{  
  "execution" : JobExecutionData,  
  "timestamp" : timestamp,  
  "clientToken" : "string"  
}
```

Donde `execution` es un [JobExecution \(p. 745\)](#) objeto. Por ejemplo:

```
{  
  "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

Obtiene información detallada acerca de una ejecución de trabajo.

Puede establecer el `jobId` para devolver la siguiente ejecución de trabajo pendiente para un objeto (con un estado de `IN_PROGRESS` o `QUEUED`).

Para invocar esta API, publique un mensaje en \$aws/things/*thingName*/jobs/*jobID*/get.

Carga de solicitud:

```
{  
  "executionNumber": long,  
  "includeJobDocument": boolean,  
  "clientToken": "string"  
}
```

thingName

El nombre del objeto asociado con el dispositivo.

jobId

El identificador único asignado a este trabajo cuando se creó.

O usa \$next para devolver la siguiente ejecución de trabajo pendiente para un objeto (con un estado de IN_PROGRESS o QUEUED). En este caso, cualquier ejecución de trabajo con estado IN_PROGRESS se devuelven primero. Las ejecuciones de trabajo se devuelven en el orden en el que se crearon.

executionNumber

(Opcional) Un número que identifica la ejecución de un trabajo en un dispositivo. Si no se especifica, se devuelve la ejecución de trabajo más reciente.

includeJobDocument

(Opcional) A menos que esté configurado en false, la respuesta contiene el documento de trabajo. El valor predeterminado es true.

El agente de mensajes publicará \$aws/things/*thingName*/jobs/*jobId*/get/accepted y \$aws/things/*thingName*/jobs/*jobId*/get/rejected incluso sin una suscripción específica a ellos. Sin embargo, para que su cliente reciba los mensajes, debe estar escuchándolos. Para obtener más información, consulte [la nota sobre los mensajes de la API de trabajos \(p. 746\)](#).

Carga de respuesta:

```
{  
  "execution" : JobExecutionData,  
  "timestamp": "timestamp",  
  "clientToken": "string"  
}
```

Donde execution es un [JobExecution \(p. 745\)](#) objeto.

UpdateJobExecution

Actualiza el estado de una ejecución de trabajo. Si lo prefiere, puede crear un temporizador de pasos estableciendo un valor para la propiedad stepTimeoutInMinutes. Si no actualiza el valor de esta propiedad ejecutando UpdateJobExecution otra vez, la ejecución del trabajo agotará el tiempo de espera cuando venza el temporizador de pasos.

Para invocar esta API, publique un mensaje en \$aws/things/*thingName*/jobs/*jobId*/update.

Carga de solicitud:

```
{  
  "status": "job-execution-state",  
  "statusDetails": {  
    "string": "string"  
    ...  
  },  
  "expectedVersion": "number",  
  "executionNumber": long,  
  "includeJobExecutionState": boolean,  
  "includeJobDocument": boolean,  
  "stepTimeoutInMinutes": long,  
  "clientToken": "string"  
}
```

status

El nuevo estado de la ejecución del trabajo (IN_PROGRESS, FAILED, SUCCEEDED, o bien REJECTED). Debe especificarse en cada actualización.

statusDetails

Un conjunto de pares nombre-valor que describen el estado de ejecución del trabajo. Si no se especifica, `statusDetails` no se modifica.

expectedVersion

La versión actual esperada de la ejecución de trabajos. Cada vez que actualiza la ejecución de trabajos, aumenta su versión. Si la versión de la ejecución del trabajo almacenada en el servicio Jobs de AWS IoT no coincide, la actualización se rechaza con un error `VersionMismatch` y se devuelve un código [ErrorResponse \(p. 730\)](#) con los datos de estado de la ejecución del trabajo actual. (Esto hace que no sea necesario realizar una solicitud `DescribeJobExecution` aparte para obtener los datos de estado de ejecución del trabajo).

executionNumber

(Opcional) Un número que identifica la ejecución de un trabajo en un dispositivo. Si no se especifica, se utiliza la ejecución de trabajo más reciente.

includeJobExecutionState

(Opcional) Cuando se incluye y establece en `true`, la respuesta contiene el `JobExecutionState`. El valor predeterminado es `false`.

includeJobDocument

(Opcional) Cuando se incluye y establece en `true`, la respuesta contiene el `JobDocument`. El valor predeterminado es `false`.

stepTimeoutInMinutes

Especifica la cantidad de tiempo que tiene este dispositivo para finalizar la ejecución de este trabajo. Si el estado de ejecución del trabajo no se ha establecido en un estado terminal antes de que este temporizador venza o antes de que se restablezca el temporizador (llamando otra vez a `UpdateJobExecution`, estableciendo el estado en `IN_PROGRESS` y especificando un valor de tiempo de espera nuevo en este campo), el estado de ejecución del trabajo se establece en `TIMED_OUT`. Configurar o restablecer este tiempo de espera no tiene ningún efecto en el tiempo de espera de la ejecución del trabajo que se pueda haber especificado cuando se creó el trabajo (utilizando `CreateJob` con el campo `timeoutConfig`).

El agente de mensajes publicará \$aws/things/`thingName`/jobs/`jobID`/update/accepted y \$aws/things/`thingName`/jobs/`jobID`/update/rejected incluso sin una suscripción específica a ellos. Sin embargo, para que su cliente reciba los mensajes, debe estar escuchándolos. Para obtener más información, consulte [la nota sobre los mensajes de la API de trabajos \(p. 746\)](#).

Carga de respuesta:

```
{  
  "executionState": JobExecutionState,  
  "jobDocument": "string",  
  "timestamp": timestamp,  
  "clientToken": "string"  
}
```

executionState

Un objeto [JobExecutionState \(p. 745\)](#).

jobDocument

Un objeto [documento de trabajo \(p. 675\)](#).

timestamp

El tiempo en segundos, desde la fecha de inicioépoca, cuando se envió el mensaje.

clientToken

Un token de cliente utilizado para correlacionar solicitudes y respuestas.

Si utiliza el protocolo MQTT, también puede realizar las siguientes actualizaciones:

JobExecutionsChanged

Se envía cuando se añade una ejecución de trabajo a la lista de ejecuciones de trabajo pendientes para un objeto, o cuando se quita de dicha lista.

Utilice el tema :

\$aws/things/*thingName*/jobs/notify

Carga de mensaje:

```
{  
  "jobs" : {  
    "JobExecutionState": [ JobExecutionSummary ... ]  
  },  
  "timestamp": timestamp  
}
```

NextJobExecutionChanged

Se envía cuando se produce un cambio en el que la ejecución de trabajo es la siguiente en la lista de ejecuciones de trabajo pendientes para un objeto, como se define para [DescribeJobExecution](#) con jobId \$next. Este mensaje no se envía cuando cambian los detalles de ejecución del siguiente trabajo, solo cuando el siguiente trabajo que devolvería [DescribeJobExecution](#) con jobId \$next ha cambiado. Considere las ejecuciones de trabajo J1 y J2 con un estado de `DEQUEUED`. J1 es el siguiente elemento de la lista de ejecuciones de trabajo pendientes. Si el estado de J2 se cambia a `IN_PROGRESS` si bien el estado de J1 permanece invariable, se envía esta notificación y contiene los detalles de J2.

Utilice el tema :

\$aws/things/*thingName*/jobs/notify-next

Carga de mensaje:

```
{  
  "execution" : JobExecution,  
  "timestamp": timestamp  
}
```

API HTTP de dispositivo de trabajo

Los dispositivos pueden comunicarse con AWS IoT Trabajos con HTTP Signature Version 4 en el puerto 443. Este es el método utilizado por el AWSSDK y CLI. Para obtener más información acerca de estas herramientas, consulte [AWS CLI Referencia de comandos: IOT-Jobs-data](#) o [AWSSDK y herramientas](#).

Los siguientes comandos están disponibles para los dispositivos que ejecutan los trabajos. Para obtener información acerca del uso de las operaciones de la API con el protocolo MQTT, consulte [API de MQTT de dispositivo de trabajo \(p. 746\)](#).

GetPendingJobExecutions

Obtiene la lista de todos los trabajos que no están en un estado terminal para un objeto específico.

HTTPS request

```
GET /things/thingName/jobs
```

Response: (Respuesta:)

```
{  
    "inProgressJobs" : [ JobExecutionSummary ... ],  
    "queuedJobs" : [ JobExecutionSummary ... ]  
}
```

Para obtener más información, 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 formato:

```
{  
    "thingName": "string"  
}
```

Para obtener más información, consulte [get-pending-job-executions](#).

StartNextPendingJobExecution

Obtiene y comienza la siguiente ejecución de trabajo pendiente para un objeto (con un estado de IN_PROGRESS o QUEUED).

- Ejecuciones de trabajos con estatus IN_PROGRESS se devuelven primero.
- Las ejecuciones de trabajo se devuelven en el orden en el que se crearon.
- Si la siguiente ejecución de trabajos pendientes es QUEUED, su estado cambia a IN_PROGRESS y los detalles de estado de la ejecución del trabajo se establecen según lo especificado.
- Si la siguiente ejecución de trabajos pendientes de inicio ya está IN_PROGRESS, sus datos de estado no cambian.
- Si no hay ejecuciones de trabajo pendientes, la respuesta no incluirá el campo execution.
- Si lo prefiere, puede crear un temporizador de pasos estableciendo un valor para el stepTimeoutInMinutes propiedad. Si no actualiza el valor de esta propiedad mediante la ejecución de UpdateJobExecution, la ejecución del trabajo agotará el tiempo de espera cuando venza el temporizador de pasos.

HTTPS request

En el siguiente ejemplo se muestra la sintaxis de la solicitud:

```
PUT /things/thingName/jobs/$next  
{  
    "statusDetails": {  
        "string": "string"  
        ...  
    },  
    "stepTimeoutInMinutes": long
```

}

Para obtener más información, consulte [StartNextPendingJobExecution](#).

CLI syntax

Sinopsis:

```
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 formato:

```
{
  "thingName": "string",
  "statusDetails": {
    "string": "string"
  },
  "stepTimeoutInMinutes": long
}
```

Para obtener más información, consulte [start-next-pending-job-execution](#).

DescribeJobExecution

Obtiene información detallada acerca de una ejecución de trabajo.

Puede establecer el `jobId` o `next` para devolver la siguiente ejecución de trabajos pendientes de un objeto. El estado de la ejecución del trabajo debe ser `QUEUED` o `IN_PROGRESS`.

HTTPS request

Solicitud:

```
GET /things/thingName/jobs/jobId?
executionNumber=executionNumber&includeJobDocument=includeJobDocument
```

Response: (Respuesta:)

```
{
  "execution" : JobExecution,
}
```

Para obtener más información, consulte [DescribeJobExecution](#).

CLI syntax

Sinopsis:

```
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 formato:

```
{  
  "jobId": "string",  
  "thingName": "string",  
  "includeJobDocument": boolean,  
  "executionNumber": long  
}
```

Para obtener más información, consulte [describe-job-execution](#).

UpdateJobExecution

Actualiza el estado de una ejecución de trabajo. Si lo prefiere, puede crear un temporizador de pasos estableciendo un valor para el `stepTimeoutInMinutes` propiedad. Si no actualiza el valor de esta propiedad ejecutando `UpdateJobExecution` otra vez, la ejecución del trabajo agotará el tiempo de espera cuando venza el temporizador de pasos.

HTTPS request

Solicitud:

```
POST /things/thingName/jobs/jobId  
{  
  "status": "job-execution-state",  
  "statusDetails": {  
    "string": "string"  
    ...  
  },  
  "expectedVersion": "number",  
  "includeJobExecutionState": boolean,  
  "includeJobDocument": boolean,  
  "stepTimeoutInMinutes": long,  
  "executionNumber": long  
}
```

Para obtener más información, consulte [UpdateJobExecution](#).

CLI syntax

Sinopsis:

```
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 formato:

```
{
```

```
"jobId": "string",
"thingName": "string",
"status": "string",
"statusDetails": {
  "string": "string"
},
"stepTimeoutInMinutes": number,
"expectedVersion": long,
"includeJobExecutionState": boolean,
"includeJobDocument": boolean,
"executionNumber": long
}
```

Para obtener más información, consulte [update-job-execution](#).

Protección de usuarios y dispositivos con AWS IoT Trabajos

Para autorizar a los usuarios a utilizar AWS IoT Trabajos con sus dispositivos, debe otorgarles permisos mediante políticas de IAM. A continuación, los dispositivos deben estar autorizados mediante AWS IoT Corepolíticas a las que conectarse de forma segura AWS IoT, recibe ejecuciones de trabajos y actualiza el estado de la ejecución.

Tipo de política de obligatorio AWS IoT Trabajos

En la tabla siguiente se muestran los distintos tipos de políticas que debe utilizar para la autorización. Para obtener más información sobre la política necesaria para utilizar, consulte [Autorización \(p. 331\)](#).

Tipo de política de obligatorio

Caso de uso	Protocolo	Autenticación	Plano de control/plano de datos	Tipo de identidad	Tipo de política de obligatorio
Autorizar a un administrador, operador o servicio en la nube a trabajar de forma segura con Jobs	HTTPS	AWS Autenticación de firma de versión 4 (puerto 443)	Plano de control y plano de datos	Amazon Cognito Identity, IAM o usuario federado	Política de IAM
Autorizar su dispositivo IoT para que funcione de forma segura con Jobs	MQTT/ HTTPS	Autenticación mutua TCP o TLS (puerto 8883 o 443)	Plano de datos	Certificados X.509	Política de AWS IoT Core

Para autorizar AWS IoT Operaciones de trabajos que se pueden realizar tanto en el plano de control como en el plano de datos, debe utilizar políticas de IAM. Las identidades deben haberse autenticado con AWS IoT para llevar a cabo estas operaciones, que deben ser [Identidades de Amazon Cognito \(p. 319\)](#) o [Usuarios, grupos y roles de IAM \(p. 318\)](#). Para obtener más información acerca de la autenticación, consulte [Autenticación \(p. 294\)](#).

Ahora, los dispositivos deben estar autorizados en el plano de datos mediante AWS IoT Core políticas para conectarse de forma segura a la puerta de enlace del dispositivo. La puerta de enlace de dispositivos permite a los dispositivos comunicarse de forma segura con AWS IoT, recibe ejecuciones de trabajos y actualiza el estado de ejecución del trabajo. La comunicación del dispositivo está protegida mediante el uso seguro [MQTT \(p. 85\)](#) o [HTTPS \(p. 95\)](#) protocolos de comunicación. Estos protocolos utilizan [Certificados de cliente X.509 \(p. 298\)](#) que son proporcionados por AWS IoT para autenticar las conexiones del dispositivo.

A continuación se muestra cómo autoriza a los usuarios, los servicios en voz alta y los dispositivos para usar AWS IoT Trabajos. Para obtener información sobre las operaciones de API de plano de control y en el plano de datos, consulte [AWS IoT API de trabajos \(p. 729\)](#).

Temas

- [Autorización de uso de usuarios y servicios en la nube AWS IoT Trabajos \(p. 756\)](#)
- [Autorización de los dispositivos para que utilicen de forma segura AWS IoT Trabajos en el plano de datos \(p. 762\)](#)

Autorización de uso de usuarios y servicios en la nube AWS IoT Trabajos

Para autorizar a sus usuarios y servicios en la nube, debe utilizar las políticas de IAM tanto en el plano de control como en el plano de datos. Las políticas deben utilizarse con el protocolo HTTPS y deben utilizar AWS Signature Versión 4 (puerto 443) para autenticar a los usuarios.

Note

AWS IoT Core no se deben utilizar políticas en el plano de control. Solo las políticas de IAM se utilizan para autorizar usuarios o servicios en la nube. Para obtener más información sobre el uso del tipo de política requerido, consulte [Tipo de política de obligatorio AWS IoT Trabajos \(p. 755\)](#).

Las políticas de IAM son documentos JSON que contienen declaraciones de políticas. Uso de las instrucciones de política `Efecto`, `Acción`, `y Recurso` para especificar los recursos, las acciones permitidas o denegadas y las condiciones en las que se permiten o deniegan las acciones. Para obtener más información, consulte [Referencia de los elementos de las políticas de JSON de IAM](#) en la guía del usuario de IAM.

Warning

Le recomendamos que no utilice permisos de comodín, como "Action": ["iot:*"] en sus políticas de IAM o AWS IoT Core políticas. El uso de permisos de comodín no es una práctica recomendada de seguridad. Para obtener más información, consulte [AWS IoT Políticas de demasiado permisivas \(p. 937\)](#).

Políticas de IAM en el plano de control

En el plano de control, las políticas de IAM utilizan el prefijo `iot:` con la acción para autorizar la operación de API de trabajos correspondiente. Por ejemplo, a la acción `iot:CreateJob` la acción de política concede permiso al usuario de para utilizar `CreateJob` API.

Acciones de política

En la tabla siguiente se muestra una lista de las acciones de la política de IAM y los permisos para utilizar las acciones de la API. Para obtener información sobre los tipos de recurso, consulte [Tipos de recurso definidos por AWS IoT](#). Para obtener más información acerca de AWS IoT acciones, consulte [Acciones definidas por AWS IoT](#).

Acciones políticas de IAM en el plano de control

Acción de política	Operación de la API	Tipos de recurso	Descripción
iot:AssociateTargetsWithJob	AssociateTargetsWithJob	tarea • thing • grupo de cosas	Representa el permiso para asociar un grupo a un trabajo continuo. Laiot:AssociateTargetsWithJobEl permiso se comprueba cada vez que se presenta una solicitud para asociar destinos.
iot:CancelJob	CancelJob	tarea	Representa el permiso para cancelar un trabajo. Laiot:CancelJobEl permiso se comprueba cada vez que se presenta una solicitud para cancelar un trabajo.
iot:CancelJobExecution	CancelJobExecution	tarea • thing	Representa el permiso para cancelar la ejecución de un trabajo. Laiot:CancelJobExecutionEl permiso se comprueba cada vez que se presenta una solicitud para cancelar la ejecución de un trabajo.
iot>CreateJob	CreateJob	tarea • thing • grupo de cosas • plantilla de trabajo	Representa el permiso para crear un trabajo. Laiot: CreateJobEl permiso se comprueba cada vez que se presenta una solicitud para crear un trabajo.
iot>CreateJobTemplate	CreateJobTemplate	tarea • plantilla de trabajo	Representa el permiso para crear una plantilla de trabajo. Laiot: CreateJobTemplateEl permiso se comprueba cada vez que se presenta una solicitud para crear una plantilla de trabajo.
iot>DeleteJob	DeleteJob	tarea	Representa el permiso para eliminar un trabajo. Laiot: DeleteJobEl permiso se comprueba cada vez que se presenta una solicitud para eliminar un trabajo.
iot>DeleteJobTemplate	DeleteJobTemplate	plantilla de trabajo	Representa el permiso para eliminar una plantilla de trabajo. Laiot: CreateJobTemplateEl permiso se comprueba cada vez que se presenta una solicitud para eliminar una plantilla de trabajo.
iot>DeleteJobExecution	DeleteJobTemplate	tarea • thing	Representa el permiso para eliminar una ejecución de trabajo. Laiot: DeleteJobExecutionEl permiso se comprueba cada vez que se presenta una solicitud para eliminar una ejecución de trabajo.
iot:DescribeJob	DescribeJob	tarea	Representa el permiso para describir un trabajo. Laiot: DescribeJobEl permiso

Acción de política	Operación de la API	Tipos de recurso	Descripción
			se comprueba cada vez que se presenta una solicitud para describir un trabajo.
iot:DescribeJobExecution	DescribeJobExecution	tarea • thing	Representa el permiso para describir una ejecución de trabajo. Laiot: DescribeJobExecutionEl permiso se comprueba cada vez que se presenta una solicitud para describir una ejecución de trabajo.
iot:DescribeJobTemplate	DescribeJobTemplate	plantilla de trabajo	Representa el permiso para describir una plantilla de trabajo. Laiot: DescribeJobTemplateEl permiso se comprueba cada vez que se presenta una solicitud para describir una plantilla de trabajo.
iot:DescribeManagedJobTemplate	DescribeManagedJobTemplate	plantilla de trabajo	Representa el permiso para describir una plantilla de trabajo. Laiot: DescribeManagedJobTemplateEl permiso se comprueba cada vez que se presenta una solicitud para describir una plantilla de trabajo.
iot:GetJobDocument	GetJobDocument	tarea	Representa el permiso para obtener el documento de trabajo de un trabajo. Laiot: GetJobDocumentEl permiso se comprueba cada vez que se presenta una solicitud para obtener un documento de trabajo.
iot>ListJobExecutionsForJob	ListJobExecutionsForJob	tareaJob	Representa el permiso para enumerar las ejecuciones de trabajo de un trabajo. Laiot: ListJobExecutionsForJobEl permiso se comprueba cada vez que se presenta una solicitud para enumerar las ejecuciones de trabajo de un trabajo.
iot>ListJobExecutionsForThing	ListJobExecutionsForThing	thing	Representa el permiso para enumerar las ejecuciones de trabajo de un trabajo. Laiot: ListJobExecutionsForThingEl permiso se comprueba cada vez que se presenta una solicitud para enumerar las ejecuciones de trabajo de un objeto.
iot>ListJobs	ListJobs	ninguno	Representa el permiso para enumerar los trabajos. Laiot: ListJobsEl permiso se comprueba cada vez que se presenta una solicitud para enumerar los trabajos.
iot>ListJobTemplates	ListJobTemplates	ninguno	Representa el permiso para enumerar las plantillas de trabajo. Laiot: ListJobTemplatesEl permiso se comprueba cada vez que se presenta una solicitud para enumerar las plantillas de trabajo.

Acción de política	Operación de la API	Tipos de recurso	Descripción
iot:ListManagedJobTemplates	ListManagedJobTemplates	ningunos	Representa el permiso para enumerar las plantillas de trabajo administradas. El permiso se comprueba cada vez que se presenta una solicitud para enumerar las plantillas de trabajo administradas.
iot:UpdateJob	UpdateJob	tarea	Representa el permiso para actualizar un trabajo. El permiso se comprueba cada vez que se presenta una solicitud para actualizar un trabajo.
iot:TagResource	TagResource	<ul style="list-style-type: none"> • tarea • plantilla de trabajo • thing 	Otorga permiso para etiquetar un recurso específico.
iot:UntagResource	UntagResource	<ul style="list-style-type: none"> • tarea • plantilla de trabajo • thing 	Otorga permiso para eliminar etiquetas en un recurso específico.

Ejemplo de política de IAM básico

A continuación se muestra un ejemplo de una política de IAM que permite al usuario el permiso para realizar las siguientes acciones para su grupo de cosas y cosas de IoT.

En el ejemplo, sustituya:

- *región* con las prácticas Región de AWS, como, por ejemplo, us-east-1.
- *account-id* con las prácticas Cuenta de AWS número, tales como 57EXAMPLE833.
- *nombre-grupo-thing* - con el nombre de tu grupo de cosas de IoT para el que estás dirigiendo trabajos, tales como FirmwareUpdateGroup.
- *nombre-cosa* con el nombre de tu cosa de IoT para la que estás apuntando a trabajos, 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 de IAM en el plano de datos

Las políticas de IAM en el plano de datos utilizan el `iotjobsdata:` prefijo para autorizar operaciones de API de trabajos que los usuarios pueden realizar. En el plano de datos, puede conceder permiso a un usuario para utilizar el `DescribeJobExecution` API mediante la acción `iotjobsdata:DescribeJobExecution` de política de.

Warning

No se recomienda utilizar políticas de IAM en el plano de datos al segmentar AWS IoT Trabajos para tus dispositivos. Recomendamos utilizar políticas de IAM en el plano de control para que los usuarios creen y administren trabajos. En el plano de datos, para autorizar a los dispositivos a recuperar ejecuciones de trabajos y actualizar el estado de ejecución, utilice [AWS IoT Core Políticas para el protocolo HTTPS \(p. 763\)](#).

Ejemplo de política de IAM básico

Las operaciones de la API que deben autorizarse las realiza un usuario que ejecute comandos de la CLI de. A continuación se muestra un ejemplo de un usuario que presenta un `DescribeJobExecution`.

En el ejemplo, sustituya:

- **región** con las prácticas Región de AWS, como, por ejemplo, `us-east-1`.
- **account-id** con las prácticas Cuenta de AWS número, tales como `57EXAMPLE833`.
- **nombre-cosa** con el nombre de tu cosa de IoT para la que estás apuntando a trabajos, como `myRegisteredThing`.
- **job-id** es el identificador único para el trabajo que se dirige mediante la 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
```

A continuación se muestra una política de IAM de ejemplo que autoriza esta acción:

```
{
    "Version": "2012-10-17",
    "Statement":
    {
        "Action": ["iotjobsdata:DescribeJobExecution"],
```

```
        "Effect": "Allow",
        "Resource": "arn:aws:iot:region:account-id:thing/thing-name",
    }
}
```

Ejemplo de política de IAM para el plano de control y en el plano de datos

Si un usuario realiza una operación de API tanto en el plano de control como en el plano de datos, la acción de política del plano de control debe utilizar `eliot:*` y la acción de la política del plano de datos debe utilizar `eliotjobsdata:*` prefijo.

Por ejemplo, a la acción `DescribeJobExecution` La API se puede utilizar tanto en el plano de control como en el plano de datos. En el plano de control, el `DescribeJobExecution` La API se utiliza para describir una ejecución de trabajo. En el plano de datos, el `DescribeJobExecution` La API se utiliza para obtener detalles de la ejecución de un trabajo.

La siguiente política de IAM autoriza a un usuario permiso para utilizar `DescribeJobExecution` API en el plano de control y en el plano de datos.

En el ejemplo, sustituya:

- `región` con las prácticas Región de AWS, como, por ejemplo, `us-east-1`.
- `account-id` con las prácticas Cuenta de AWS número, tales como `57EXAMPLE833`.
- `nombre-cosa` con el nombre de tu cosa de IoT para la que estás apuntando a trabajos, 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 el etiquetado de recursos de IoT

Para obtener un mejor control sobre los trabajos y las plantillas de trabajos que un usuario puede crear, modificar o utilizar, puede adjuntar etiquetas a los trabajos o plantillas de trabajos. Las etiquetas también le ayudan a discernir la propiedad y asignar y asignar costos colocándolos en grupos de facturación y adjuntando etiquetas a ellos.

Cuando un usuario desea etiquetar sus trabajos o plantillas de trabajo que ha creado mediante el AWS Management Console o el AWS CLI, su política de IAM debe conceder permisos de usuario para etiquetarlos. Para conceder permisos, la política de IAM debe utilizar `eliot:TagResource` acción.

Para obtener información general sobre el etiquetado de los recursos, consulte [Etiquetado de los recursos de AWS IoT \(p. 288\)](#).

Ejemplo de política de IAM

Para ver un ejemplo que muestra cómo conceder permisos de etiquetado, considere un usuario que ejecuta el siguiente comando para crear un trabajo y etiquetarlo en un entorno específico.

En el ejemplo, sustituya:

- *región* con las prácticas Región de AWS, como, por ejemplo, `us-east-1`.
- *account-id* con las prácticas Cuenta de AWS número, tales como `57EXAMPLE833`.
- *nombre-cosa* con el nombre de tu cosa de IoT para la que estás apuntando a trabajos, 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 ejemplo, debe utilizar la siguiente política de 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/*"
        ]
    }
}
```

Autorización de los dispositivos para que utilicen de forma segura AWS IoT Trabajos en el plano de datos

Para autorizar a los dispositivos a interactuar de forma segura con AWS IoT Trabajos en el plano de datos, debe utilizar AWS IoT Core políticas. AWS IoT Core políticas para trabajos son documentos JSON que contienen declaraciones de políticas. Estas políticas también utilizan Efecto, Acción, y Recurso elementos y seguir una convención similar a las políticas de IAM. Para obtener más información acerca de los elementos, consulte [Referencia de los elementos de políticas de JSON de IAM](#) en la [IAM User Guide](#).

Las políticas se pueden utilizar con protocolos MQTT y HTTPS y deben utilizar la autenticación mutua TCP o TLS para autenticar los dispositivos. A continuación se muestra cómo utilizar estas políticas en los distintos protocolos de comunicación.

Warning

Le recomendamos que no utilice permisos de comodín, como `"Action": ["iot:"]` en sus políticas de IAM o AWS IoT Core políticas. El uso de permisos de comodín no es una práctica recomendada de seguridad. Para obtener más información, consulte [AWS IoT Políticas de demasiado permisivas \(p. 937\)](#).

AWS IoT CorePolíticas para el protocolo MQTT

AWS IoT Core las políticas para el protocolo MQTT le otorgan permisos para utilizar las acciones de la API MQTT del dispositivo de trabajos. Las operaciones de la API de MQTT se utilizan para trabajar con temas de MQTT reservados para los comandos de trabajos. Para obtener más información sobre estas operaciones de API, consulte [API de MQTT de dispositivo de trabajo \(p. 746\)](#).

Las políticas de MQTT utilizan acciones políticas tales como `iot:Connect`, `iot:Publish`, `iot:Subscribe`, `yiot:Receieve` para trabajar con los temas de los trabajos. Estas políticas le permiten conectarse con el agente de mensajes de suscribirse a los temas MQTT de trabajos y enviar y recibir mensajes MQTT entre sus dispositivos y la nube. Para obtener más información sobre estas acciones, consulte [Acciones de política de AWS IoT Core \(p. 334\)](#).

Para obtener información sobre los temas de AWS IoT Trabajos, consulte [Temas de trabajos \(p. 107\)](#).

Ejemplo de política básica de MQTT

El siguiente ejemplo muestra cómo utilizar `iot:Publish` y `iot:Subscribe` para publicar y suscribirse a puestos de trabajo y ejecuciones de trabajos.

En el ejemplo, sustituya:

- `región` con las prácticas Región de AWS, como, por ejemplo, `us-east-1`.
- `account-id` con las prácticas Cuenta de AWS número, tales como `57EXAMPLE833`.
- `nombre-cosa` con el nombre de tu cosa de IoT para la que estás apuntando a trabajos, 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 el protocolo HTTPS

AWS IoT Core las directivas del plano de datos también pueden utilizar el protocolo HTTPS con el mecanismo de autenticación TLS para autorizar los dispositivos. En el plano de datos, las políticas utilizan el prefijo `iot:jobsdata` para autorizar operaciones de API de trabajos que sus dispositivos pueden realizar. Por ejemplo, a la acción `iot:jobsdata:DescribeJobExecution` La acción de política concede permiso al usuario de para utilizar `DescribeJobExecution` API.

Note

Las acciones de política del plano de datos deben utilizar el prefijo `iot:jobsdata`. En el plano de control, las acciones deben utilizar el prefijo `iot:`. Para ver un ejemplo de política de IAM cuando

se utilizan acciones de política de plano de control y plano de datos, consulte [Ejemplo de política de IAM para el plano de control y en el plano de datos \(p. 761\)](#).

Acciones de política

En la tabla siguiente se muestra una lista de AWS IoT Core acciones de políticas y permisos para autorizar a los dispositivos a utilizar las acciones de la API. Para ver la lista de operaciones de la API que puede realizar en el plano de datos de, consulte [API HTTP de dispositivo de trabajo \(p. 751\)](#).

Note

Estas acciones de política de ejecución de trabajo se aplican únicamente al punto de enlace HTTP TLS. Si utiliza el punto de enlace MQTT, debe utilizar las acciones de política MQTT definidas anteriormente.

AWS IoT Core acciones de políticas en el plano de datos

Acción de política	Operación de la API	Tipos de recurso	Descripción
iotjobsdata:DescribeJobExecution	DescribeJobExecution	tarea • thing	Representa el permiso para recuperar una ejecución de trabajo. La acción <code>iotjobsdata:DescribeJobExecution</code> se comprueba cada vez que se presenta una solicitud para recuperar la ejecución de un trabajo.
iotjobsdata:GetPendingJobExecutions	GetPendingJobExecutions	thing	Representa el permiso para recuperar la lista de trabajos que no están en un estado final para un objeto. El permiso <code>iotjobsdata:GetPendingJobExecutions</code> se comprueba cada vez que se presenta una solicitud para recuperar la lista.
iotjobsdata:StartNextPendingJobExecution	StartNextPendingJobExecution	thing	Representa el permiso para obtener e iniciar la próxima ejecución de trabajo pendiente para un objeto. Es decir, para actualizar una ejecución de trabajo con un estado <code>QUEUED</code> o <code>IN_PROGRESS</code> . El permiso <code>iot:StartNextPendingJobExecution</code> se comprueba cada vez que se presenta una solicitud para iniciar la siguiente ejecución de trabajo pendiente.
iotjobsdata:UpdateJobExecution	UpdateJobExecution	thing	Representa el permiso para actualizar una ejecución de trabajo. El permiso <code>iot:UpdateJobExecution</code> se comprueba cada vez que se presenta una solicitud para actualizar el estado de una ejecución de trabajo.

Ejemplo de política de básico

A continuación se muestra un ejemplo de AWS IoT Core que permite al usuario permiso para realizar las acciones en las operaciones de la API del plano de datos de cualquier recurso. Puede aplicar el ámbito de la política a un recurso específico, como un elemento de IoT. En el ejemplo, reemplaza:

- *región* con las prácticasRegión de AWS tales como us-east-1.
- *account-id* con las prácticasCuenta de AWS número, tales como 57EXAMPLE833.
- *nombre-cosa* con el nombre del objeto de 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"  
        }  
    ]  
}
```

Un ejemplo de cuándo debe utilizar estas políticas puede ser cuando los dispositivos IoT utilizan un AWS IoT Core política para acceder a una de estas operaciones de API, como el siguiente ejemplo de `DescribeJobExecution` API:

```
GET /things/thingName/jobs/jobId?  
executionNumber=executionNumber&includeJobDocument=includeJobDocument&namespaceId=namespaceId  
HTTP/1.1
```

Límites de los trabajos

Para obtener información sobre los límites de trabajos, consulte [AWS IoT Quotas y puntos de enlace de administración de dispositivos](#) en la [AWS Reference general de](#).

Tunelización segura de AWS IoT

Cuando los dispositivos se implementan detrás de firewalls restringidos en sitios remotos, necesita una forma de obtener acceso a esos dispositivos para solucionar problemas, actualizar la configuración y otras tareas operativas. Utilice túnel seguro para establecer una comunicación bidireccional con dispositivos remotos a través de una conexión segura administrada por AWS IoT. La tunelización segura no requiere actualizaciones de las reglas de firewall entrante existentes, por lo que puede mantener el mismo nivel de seguridad proporcionado por las reglas de firewall en un sitio remoto.

Por ejemplo, un sensor instalado en una fábrica que está a varios kilómetros de distancia está teniendo problemas para medir la temperatura de la fábrica. Puede utilizar la tunelización segura para abrir e iniciar rápidamente una sesión en ese sensor. Después de identificar el problema (por ejemplo, un archivo de configuración incorrecto), puede restablecer el archivo y reiniciar el sensor a través de la misma sesión. En comparación con una solución de problemas más tradicional (por ejemplo, enviar a un técnico a la fábrica para que revise el sensor), la tunelización segura reduce la respuesta a incidentes y el tiempo de recuperación, así como los costos de explotación.

¿Qué es el túnel seguro?

Utilice túneles seguros para acceder a los dispositivos que se implementan detrás de firewalls restringidos por puertos en sitios remotos. Puede conectarse al dispositivo de destino desde su portátil o equipo de escritorio como dispositivo de origen mediante elNube de AWS. El origen y destino se comunican mediante un proxy local de código abierto que se ejecuta en cada dispositivo. El proxy local se comunica con elNube de AWS mediante un puerto abierto permitido por el cortafuegos, normalmente 443. Los datos que se transmiten a través del túnel se cifran con Transport Layer Security (TLS).

Temas

- [Conceptos de tunelización segura \(p. 766\)](#)
- [Cómo funciona el tunelización segura de \(p. 767\)](#)
- [Ciclo de vida del túnel seguro \(p. 768\)](#)

Conceptos de tunelización segura

El túnel seguro utiliza los siguientes términos al establecer la comunicación con dispositivos remotos. Para obtener información sobre cómo funciona el tunelización segura de, consulte [Cómo funciona el tunelización segura de \(p. 767\)](#).

Token de acceso de cliente (CAT)

Un par de tokens generados por la tunelización segura cuando se crea un nuevo túnel. Los dispositivos de origen y destino utilizan el CAT para conectarse al servicio de túnel seguro.

Token de cliente

Un valor único generado por el cliente que AWS IoT túnel seguro se puede utilizar para todas las conexiones de reintento posteriores al mismo túnel. Este campo es opcional. Si no se proporciona el token de cliente, el token de acceso de cliente (CAT) solo se puede utilizar una vez para el mismo túnel. Se rechazarán los intentos de conexión posteriores que utilicen el mismo CAT.

Aplicación de destino

La aplicación que se ejecuta en el dispositivo de destino. Por ejemplo, la aplicación de destino puede ser un daemon SSH para establecer una sesión SSH mediante tunelización segura.

Dispositivo de destino

El dispositivo remoto al que desea obtener acceso.

Agente de dispositivo

Una aplicación de IoT que se conecta al gateway de dispositivos de AWS IoT y está a la escucha de nuevas notificaciones de túnel a través de MQTT. Para obtener más información, consulte [Fragmento de agente de IoT \(p. 784\)](#).

Proxy local

Proxy de software que se ejecuta en los dispositivos de origen y destino y transmite un flujo de datos entre la tunelización segura y la aplicación del dispositivo. El proxy local se puede ejecutar en modo de origen o modo de destino. Para obtener más información, consulte [Proxy local \(p. 771\)](#).

Dispositivo de origen

El dispositivo que un operador utiliza para iniciar una sesión en el dispositivo de destino, normalmente un equipo portátil o de sobremesa.

Túnel

Una ruta lógica a través de AWS IoT que permite la comunicación bidireccional entre un dispositivo de origen y un dispositivo de destino.

Cómo funciona el tunelización segura de

A continuación se muestra cómo el túnel seguro establece una conexión entre el dispositivo de origen y de destino. Para obtener información sobre los diferentes términos, como el token de acceso de cliente (CAT), consulte [Conceptos de tunelización segura \(p. 766\)](#).

1. Abrir un túnel

Para abrir un túnel para iniciar una sesión con el dispositivo de destino remoto, puede utilizar la AWS Management Console, el [AWS CLI](#) túnel abierto o el [API de OpenTunnel](#).

2. Descargar el par de tokens de acceso de clientes

Después de abrir un túnel, puede descargar el token de acceso de cliente (CAT) para su origen y destino y guardarlo en su dispositivo de origen. Debe recuperar el CAT y guardarlo ahora antes de iniciar el proxy local.

3. Iniciar proxy local en modo de destino

El agente de IoT que se ha instalado y se ejecuta en su dispositivo de destino se suscribirá al tema `MQTT reservado.$aws/things/<thing-name>/tunnels/notify` y recibirá el CAT. Aquí, `nombre-costa` es el nombre de la AWS IoT cosa que creas para tu destino. Para obtener más información, consulte [Temas de tunelización segura \(p. 113\)](#).

A continuación, el agente de IoT utiliza el CAT para iniciar el proxy local en modo de destino y configurar una conexión en el lado de destino del túnel. Para obtener más información, consulte [Fragmento de agente de IoT \(p. 784\)](#).

4. Iniciar proxy local en modo fuente

Después de abrir el túnel, AWS IoT Device Management proporciona el CAT del origen que puede descargar en el dispositivo de origen. Puede utilizar el CAT para iniciar el proxy local en modo de origen, que luego conecta el lado de origen del túnel. Para obtener más información sobre el proxy local, consulte [Proxy local \(p. 771\)](#).

5. Abrir una sesión SSH

Dado que ambos lados del túnel están conectados, puede iniciar una sesión SSH utilizando el proxy local del lado fuente.

Para obtener más información sobre cómo utilizar el AWS Management Console para abrir un túnel e iniciar una sesión SSH, consulte [Abra un túnel e inicie la sesión SSH en un dispositivo remoto \(p. 768\)](#).

En el siguiente video se describe cómo funciona el túnel seguro y lo guía por el proceso de configuración de una sesión SSH en un dispositivo Raspberry Pi.

Ciclo de vida del túnel seguro

Los túneles pueden tener el estado OPEN, CLOSED, CONNECTED o DISCONNECTED. Las conexiones al túnel pueden tener el estado CONNECTED o DISCONNECTED. A continuación se muestra cómo funcionan los distintos estados de túnel y conexión.

1. Cuando se abre un túnel, este tiene el estado OPEN. El estado de conexión de origen y destino del túnel se establece en DISCONNECTED.
2. Cuando un dispositivo (de origen o destino) se conecta al túnel, el estado de conexión correspondiente cambia a CONNECTED.
3. Cuando un dispositivo se desconecta del túnel mientras permanece el estado del túnel OPEN, el estado de conexión correspondiente cambia de nuevo a DISCONNECTED. Un dispositivo puede conectarse y desconectarse de un túnel varias veces mientras el túnel tenga el estado OPEN.
4. Cuando llamas CloseTunnel al túnel permanece OPEN durante más tiempo que el MaxLifetimeTimeout valor, el estado de un túnel se convierte a CLOSED. Puede configurar MaxLifetimeTimeout al llamar a OpenTunnel. MaxLifetimeTimeout está establecido de forma predeterminada en 12 horas si no se especifica un valor.

Note

Un túnel no se puede volver a abrir cuando tiene el estado CLOSED.

5. Puedes llamar DescribeTunnel y ListTunnels para ver los metadatos de túnel mientras el túnel está visible. El túnel se puede ver en el AWS IoT consola de durante al menos tres horas después de eliminarla.

AWS IoT tutoriales de tunelización segura

AWS IoT La tunelización segura ayuda a los clientes a establecer una comunicación bidireccional con dispositivos remotos que están detrás del firewall a través de una conexión segura administrada por AWS IoT. Los siguientes tutoriales le ayudarán a aprender a comenzar a utilizar tunelización segura.

Para realizar una demostración rápida AWS IoT túnel seguro, utilice nuestra [AWS IoT demostración de túnel seguro en GitHub](#).

AWS IoT tutoriales de tunelización segura

- [Abra un túnel e inicie la sesión SSH en un dispositivo remoto \(p. 768\)](#)

Abra un túnel e inicie la sesión SSH en un dispositivo remoto

En este tutorial, abrirá un túnel y lo utilizará para iniciar una sesión SSH en un dispositivo remoto. El dispositivo remoto está detrás de firewalls que bloquean todo el tráfico entrante, lo que imposibilita una

conexión SSH directa en el dispositivo. Antes de comenzar, asegúrese de que sabe cómo [registrar un dispositivo en el registro de AWS IoT](#) y [conectar un dispositivo al gateway de dispositivos de AWS IoT](#).

Requisitos previos

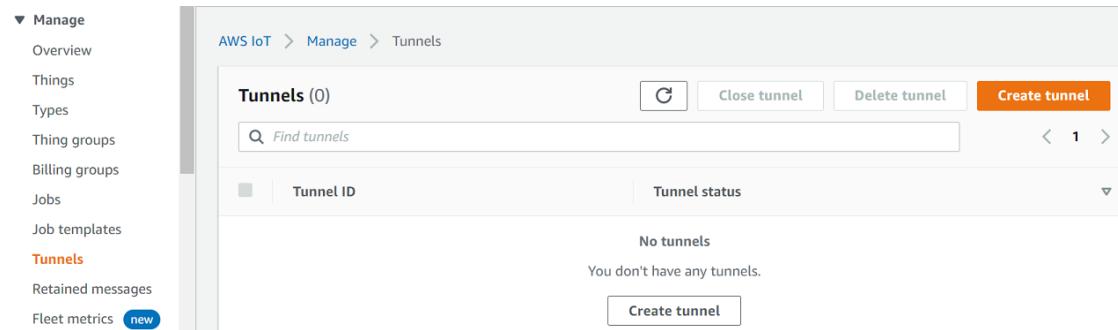
- Los firewalls detrás del dispositivo remoto deben permitir el tráfico saliente en el puerto 443.
- Ha creado una objeto de IoT denominado `RemoteDeviceA` en el registro de AWS IoT.
- Tiene un agente de dispositivo de IoT ejecutándose en el dispositivo remoto que se conecta al gateway de dispositivos de AWS IoT y está configurado con una suscripción a un tema de MQTT. Este tutorial incluye un fragmento que muestra cómo implementar un agente. Para obtener más información, consulte [Fragmento de agente de IoT \(p. 784\)](#).
- Debe tener un daemon SSH ejecutándose en el dispositivo remoto.
- Ha descargado el código fuente del proxy local de [GitHub](#) y lo ha compilado para la plataforma de su elección. Nos referiremos al archivo ejecutable del proxy local compilado como `localproxy` en este tutorial.

Abrir un túnel

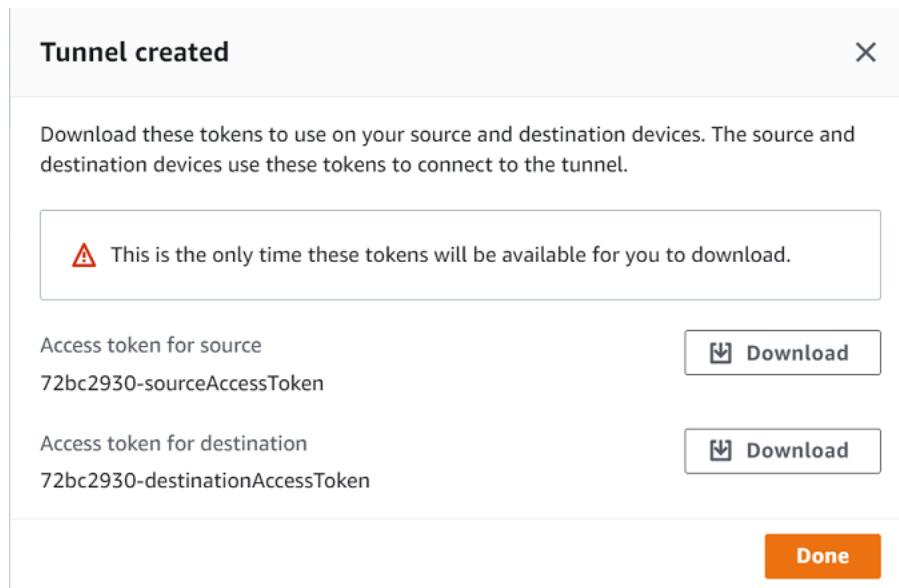
Si configura el destino al llamar `OpenTunnel`, la tunelización segura entrega el token de acceso del cliente de destino al dispositivo remoto a través de MQTT y el tema reservado de MQTT (`$aws/things/RemoteDeviceA/tunnels/notify`). Para obtener más información, consulte [Temas reservados \(p. 100\)](#). Al recibir el mensaje de MQTT, el agente IoT del dispositivo remoto inicia el proxy local en modo de destino. Puede omitir la configuración de destino si desea entregar el token de acceso de cliente de destino al dispositivo remoto a través de otro método. Para obtener más información, consulte [Configuración de un dispositivo remoto \(p. 790\)](#).

Para abrir un túnel en la consola

1. En el navegador [AWS IoT consola](#), vaya a `Manage` y elige `Tunelización`.



2. Elegir `Crear túnel`.
3. Introduzca una descripción de túnel, el nombre de la cosa para la que desea abrir un túnel, el servicio que se va a utilizar, como SSH o FTP, la duración del tiempo de espera del túnel y las etiquetas de recursos como pares clave-valor para ayudarle a identificar el recurso.
4. Descargue los tokens de acceso de cliente para el origen y el destino. Los tokens no estarán disponibles para descargar una vez que elijas `Terminado`.



5. Seleccione Done (Listo).

Iniciar el proxy local

Abra un terminal en su portátil, copie el token de acceso de cliente de origen y úselo para iniciar el proxy local en modo de origen. En el siguiente comando, el proxy local está configurado para atender nuevas conexiones en el puerto 5555.

```
./localproxy -r us-east-1 -s 5555 -t source-client-access-token
```

Note

La Región de AWSEn este comando debe ser el mismoRegión de AWSdonde se creó el túnel.

-r

Especifica elRegión de AWSdonde se crea el túnel.

-s

Especifica el puerto al que debe conectarse el proxy.

-t

Especifica el texto del token del cliente.

Note

Si recibe el siguiente error, configure la ruta de acceso de entidad de certificación. Para obtener información, consulte [GitHub](#).

```
Could not perform SSL handshake with proxy server: certificate verify failed
```

Iniciar una sesión SSH

Abra otro terminal y utilice el siguiente comando para iniciar una nueva sesión SSH conectándose al proxy local en el puerto 5555.

```
ssh username@localhost -p 5555
```

Es posible que se pida una contraseña para la sesión SSH. Cuando haya terminado con la sesión SSH, escriba **exit** para cerrar la sesión.

Cierre del túnel

1. Abra la [consola de AWS IoT](#).
2. Elija el túnel y, en Actions (Acciones), elija Close (Cerrar). Al cerrar el túnel, se cierran las dos instancias del proxy local.

Proxy local

El proxy local transmite los datos enviados por la aplicación que se ejecuta en el dispositivo de origen mediante túneles seguros a través de un WebSocket conexión segura. Puede descargar el proxy local de origen en [GitHub](#).

El proxy local puede ejecutarse en dos modos: `source` o `destination`. En el modo de origen, el proxy local se ejecuta en el mismo dispositivo o red que la aplicación cliente que inicia la conexión TCP. En el modo de destino, el proxy local se ejecuta en el dispositivo remoto, junto con la aplicación de destino. Un único túnel puede admitir hasta tres conexiones TCP a la vez mediante la multiplexación de túneles. Para obtener más información, consulte [Flujos de datos multiplex en un túnel seguro \(p. 780\)](#).

Cómo utilizar el proxy local

Puede ejecutar el proxy local en los dispositivos de origen y de destino para transmitir datos a los extremos de túnel seguro. Si los dispositivos se encuentran en una red que utiliza un proxy web, el proxy web puede interceptar las conexiones antes de reenviarlas a Internet. En este caso, tendrá que configurar el proxy local para que use el proxy web. Para obtener más información, consulte [Configurar proxy local para dispositivos que utilizan proxy web \(p. 775\)](#).

Flujo de trabajo proxy local

En los siguientes pasos se muestra cómo se ejecuta el proxy local en los dispositivos de origen y de destino.

1. Connect el proxy local a un túnel seguro

En primer lugar, el proxy local debe establecer una conexión para un túnel seguro. Cuando inicie el proxy local, utilice los argumentos siguientes:

- La `-r`argumento para especificar el `Region` de AWS en la que se abre el túnel.
- La `-t`para pasar el token de acceso de cliente de origen o destino devuelto desde `openTunnel`.

Note

Dos proxies locales que utilicen el mismo valor de token de acceso de cliente no se pueden conectar al mismo tiempo.

2. Realizar acciones de origen o destino

Después de la WebSocket Se establece la conexión, el proxy local realiza acciones en modo de origen o en modo de destino, según su configuración.

De forma predeterminada, el proxy local intenta volver a conectarse a un túnel seguro si se producen errores de entrada/salida (E/S) o si el WebSocket la conexión se cierra inesperadamente. Esto hace que la conexión TCP se cierre. Si se produce algún error de socket TCP, el proxy local envía

un mensaje a través del túnel para notificar al otro extremo que cierre su conexión TCP. De forma predeterminada, el proxy local siempre usa la comunicación SSL.

3. Detener el proxy local

Después de utilizar el túnel, puede detener el proceso del proxy local sin problemas. Le recomendamos que cierre explícitamente el túnel llamando a `CloseTunnel`. Es posible que los clientes del túnel activos no se cierren inmediatamente después de llamar a `CloseTunnel`.

Para obtener más información sobre cómo utilizar el AWS Management Console para abrir un túnel e iniciar una sesión SSH, consulte [Abra un túnel e inicie la sesión SSH en un dispositivo remoto \(p. 768\)](#).

Prácticas recomendadas para proxy local

Cuando ejecute el proxy local, siga estas prácticas recomendadas:

- Evite el uso del argumento `-t` del proxy local para pasar un token de acceso. Se recomienda utilizar la variable de entorno `AWSIOT_TUNNEL_ACCESS_TOKEN` para establecer el token de acceso del proxy local.
- Ejecute el ejecutable del proxy local con privilegios mínimos en el sistema operativo o en el entorno.
 - Evite ejecutar el proxy local como administrador en Windows.
 - Evite ejecutar el proxy local como raíz en Linux y macOS.
- Considere la posibilidad de ejecutar el proxy local en hosts distintos, contenedores, entornos de pruebas, chroot jail o en un entorno virtualizado.
- Cree el proxy local con indicadores de seguridad relevantes, en función de su cadena de herramientas.
- En dispositivos con varias interfaces de red, utilice el argumento `-b` para enlazar el socket TCP a la interfaz de red utilizada para comunicarse con la aplicación de destino.

Comando y salida de ejemplo

A continuación se muestra un ejemplo de un comando que ejecuta en un SO Linux y el resultado correspondiente. En el ejemplo se muestra un proxy web que escucha en un puerto HTTP y cómo se puede configurar el proxy local en ambos `sourceydestination` Modos. Para poder ejecutar estos comandos, debe haber abierto un túnel y haber obtenido los tokens de acceso del cliente para el origen y el destino. También debe haber creado el proxy local tal y como se ha descrito anteriormente.

El proxy local actualiza el protocolo HTTPS a WebSockets para establecer una conexión de larga duración y, a continuación, comienza a transmitir datos a través de la conexión a los extremos del dispositivo de túnel seguro.

Note

Los siguientes comandos utilizados en los ejemplos utilizan la `verbosity` para ilustrar una descripción general de los distintos pasos descritos anteriormente después de ejecutar el proxy local. Le recomendamos que utilice este indicador solo para fines de pruebas.

Ejecución de proxy local en modo fuente

Los siguientes comandos muestran cómo ejecutar el proxy local en modo de origen.

Linux/macOS

En Linux o macOS, ejecute los siguientes comandos en el terminal para configurar e iniciar el proxy local en el origen.

```
export AWSIOT_TUNNEL_ACCESS_TOKEN=${access_token}
```

```
./localproxy -s 5555 -v 5 -r us-west-2
```

Donde:

- **-s**es el puerto de escucha de origen, que inicia el proxy local en modo fuente.
- **-v**es la verbosidad de la salida, que puede ser un valor entre cero y seis.
- **-r**es la región del extremo en la que se abre el túnel.

Para obtener más información sobre los parámetros, consulte[Opciones definidas mediante argumentos de línea de comandos](#).

Windows

En Windows, configura el proxy local de forma similar al que lo hace para Linux o macOS, pero la forma en que define las variables de entorno es diferente de las demás plataformas. Ejecute los comandos siguientes en elcmdventana para configurar e iniciar el proxy local en su origen.

```
set AWSIOT_TUNNEL_ACCESS_TOKEN=${access_token}  
.\\localproxy -s 5555 -v 5 -r us-west-2
```

Donde:

- **-s**es el puerto de escucha de origen, que inicia el proxy local en modo fuente.
- **-v**es la verbosidad de la salida, que puede ser un valor entre cero y seis.
- **-r**es la región del extremo en la que se abre el túnel.

Para obtener más información sobre los parámetros, consulte[Opciones definidas mediante argumentos de línea de comandos](#).

A continuación se muestra un ejemplo de resultado de cómo ejecutar el proxy local ensourceModo.

```
...  
  
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
```

Ejecución de proxy local en modo de destino

Los siguientes comandos muestran cómo ejecutar el proxy local en modo de destino.

Linux/macOS

En Linux o macOS, ejecute los siguientes comandos en el terminal para configurar e iniciar el proxy local en su destino.

```
export AWSIOT_TUNNEL_ACCESS_TOKEN=${access_token}
./localproxy -d 22 -v 5 -r us-west-2
```

Donde:

- **-d**es la aplicación de destino que inicia el proxy local en modo de destino.
- **-v**es la verbosidad de la salida, que puede ser un valor entre cero y seis.
- **-r**es la región del extremo en la que se abre el túnel.

Para obtener más información sobre los parámetros, consulte[Opciones definidas mediante argumentos de línea de comandos](#).

Windows

En Windows, configura el proxy local de forma similar al que lo hace para Linux o macOS, pero la forma en que define las variables de entorno es diferente de las demás plataformas. Ejecute los comandos siguientes en elcmdventana para configurar e iniciar el proxy local en su destino.

```
set AWSIOT_TUNNEL_ACCESS_TOKEN=${access_token}
.\localproxy -d 22 -v 5 -r us-west-2
```

Donde:

- **-d**es la aplicación de destino que inicia el proxy local en modo de destino.
- **-v**es la verbosidad de la salida, que puede ser un valor entre cero y seis.
- **-r**es la región del extremo en la que se abre el túnel.

Para obtener más información sobre los parámetros, consulte[Opciones definidas mediante argumentos de línea de comandos](#).

A continuación se muestra un ejemplo de resultado de cómo ejecutar el proxy local en `destinationModo`.

```
...
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 utilizan proxy web

Puede utilizar el proxy local en AWS IoT dispositivos con los que comunicarse AWS IoT API de tunelización segura. El proxy local transmite los datos enviados por la aplicación del dispositivo mediante túnel seguro a través de un WebSocket conexión segura. El proxy local puede funcionar en `source` o `destination` modo. En `source` modo, se ejecuta en el mismo dispositivo o red que inicia la conexión TCP. En `destination` modo, el proxy local se ejecuta en el dispositivo remoto, junto con la aplicación de destino. Para obtener más información, consulte [Proxy local \(p. 771\)](#).

El proxy local necesita conectarse directamente a Internet para utilizarlo AWS IoT tunelización segura. Para una conexión TCP de larga duración con túnel seguro, el proxy local actualiza la solicitud HTTPS para establecer un WebSockets conexión a uno de los [puntos finales de conexión de dispositivos de túnel seguro](#).

Si tus dispositivos están en una red que utiliza un proxy web, el proxy web puede interceptar las conexiones antes de reenviarlas a Internet. Para establecer una conexión de larga duración a los extremos de conexión de dispositivos de túnel seguro, configure el proxy local para que utilice el proxy web tal y como se describe en la [especificación de websocket](#).

Note

La [AWS IoT Cliente de dispositivos \(p. 1277\)](#) no admite dispositivos que utilizan un proxy web. Para trabajar con el proxy web, deberá utilizar un proxy local y configurarlo para que funcione con un proxy web como se describe a continuación.

En los siguientes pasos se muestra cómo funciona el proxy local con un proxy web.

1. El proxy local envía un `HTTP/1.1 CONNECT` solicitud al proxy web que contiene la dirección remota del servicio de túnel seguro, junto con la información de autenticación de proxy web.
2. El proxy web creará entonces una conexión de larga duración a los extremos de túnel seguro remoto.
3. La conexión TCP está establecida y el proxy local funcionará ahora en los modos de origen y destino para la transmisión de datos.

Para completar este procedimiento, siga estos pasos.

- [Compilación del proxy local \(p. 776\)](#)

- [Configurar el proxy web \(p. 776\)](#)
- [Configure e inicie el proxy local \(p. 777\)](#)

Compilación del proxy local

Abra el icono[código fuente de proxy local](#)en la GitHub repositorio y siga las instrucciones para crear e instalar el proxy local.

Configurar el proxy web

El proxy local se basa en el mecanismo de túnel HTTP descrito por el[Especificación HTTP/1.1](#). Para cumplir con las especificaciones, su proxy web debe permitir que los dispositivos utilicen elCONNECTmétodo.

La forma de configurar el proxy web depende del proxy web que esté utilizando y de la versión del proxy web. Para asegurarse de configurar correctamente el proxy web, consulte la documentación de su proxy web.

Para configurar el proxy web, primero identifique la URL del proxy web y confirme si su proxy web admite la túnel HTTP. La URL del proxy web se utilizará más adelante cuando configure e inicie el proxy local.

1. Identifica la URL de tu proxy web

La URL de proxy web tendrá el siguiente formato.

```
protocol://web_proxy_host_domain:web_proxy_port
```

AWS IoT túnel seguro solo admite la autenticación básica para proxy web. Para utilizar la autenticación básica, debe especificar la[username](#)y[password](#)como parte de la URL del proxy web. La URL del proxy web tendrá el siguiente formato.

```
protocol://username:password@web_proxy_host_domain:web_proxy_port
```

- *protocolo*puede serhttp ohttps. Le recomendamos que utilice https.
- *web_proxy_host_domain*es la dirección IP de su proxy web o un nombre de DNS que se resuelve en la dirección IP de su proxy web.
- *web_proxy_port*es el puerto en el que el proxy web escucha.
- El proxy web utiliza estou[username](#)y[password](#)para autenticar la solicitud.

2. Pruebe la URL de proxy web

Para confirmar si el proxy web admite el túnel TCP, utilice uncurl y asegúrese de obtener un2xxx o un3xxx respuesta.

Por ejemplo, si la URL de su proxy web eshttps://server.com:1235, usa unproxy-insecure marca con elcurl porque el proxy web puede depender de un certificado autofirmado.

```
export HTTPS_PROXY=https://server.com:1235
curl -I https://aws.amazon.com --proxy-insecure
```

Si la URL de tu proxy web tiene unhttp port (por ejemplo,http://server.com:1234), no es necesario que utilice elproxy-insecure indicador.

```
export HTTPS_PROXY=http://server.com:1234
curl -I https://aws.amazon.com
```

Configure e inicie el proxy local

Para configurar el proxy local para que use un proxy web, debe configurar el `HTTPS_PROXY` variable de entorno con los nombres de dominio de DNS o con las direcciones IP y los números de puerto que utiliza el proxy web.

Después de configurar el proxy local, puede utilizar el proxy local tal y como se explica en este [README](#) no válido.

Note

La declaración de variable de entorno distingue entre mayúsculas y minúsculas. Le recomendamos que defina cada variable una vez utilizando todas las letras en mayúscula o en minúscula. En los ejemplos siguientes, se muestra la variable de entorno declarada en mayúscula. Si se especifica la misma variable con letras mayúsculas y minúsculas, prevalece la variable especificada con letras minúsculas.

Los siguientes comandos muestran cómo configurar el proxy local que se ejecuta en su destino para utilizar el proxy web e iniciar el proxy local.

- `AWSIOT_TUNNEL_ACCESS_TOKEN`: Esta variable contiene el token de acceso de cliente (CAT) del destino.
- `HTTPS_PROXY`: Esta variable contiene la URL del proxy web o la dirección IP para configurar el proxy local.

Los comandos que se muestran en los ejemplos siguientes dependen del sistema operativo que utilice y de si el proxy web escucha en un puerto HTTP o HTTPS.

Reproducción de proxy web en un puerto HTTP

Si su proxy web escucha en un puerto HTTP, puede proporcionar la dirección IP o la URL del proxy web para el `HTTPS_PROXY` variable.

Linux/macOS

En Linux o macOS, ejecute los siguientes comandos en el terminal para configurar e iniciar el proxy local en su destino y utilizar un proxy web que escucha en un puerto HTTP.

```
export AWSIOT_TUNNEL_ACCESS_TOKEN=${access_token}
export HTTPS_PROXY=http://proxy.example.com:1234
./localproxy -r us-east-1 -d 22
```

Si tiene que autenticarse con el proxy, debe especificar un `username` y `password` como parte de la `HTTPS_PROXY` variable.

```
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

En Windows, configura el proxy local de forma similar al que lo hace para Linux o macOS, pero la forma en que define las variables de entorno es diferente de las demás plataformas. Ejecute los comandos siguientes en el `cmd` para configurar e iniciar el proxy local en su destino para utilizar un proxy web que escucha en un puerto HTTP.

```
set AWSIOT_TUNNEL_ACCESS_TOKEN=${access_token}
```

```
set HTTPS_PROXY=http://proxy.example.com:1234
.\localproxy -r us-east-1 -d 22
```

Si tiene que autenticarse con el proxy, debe especificar un **username** y **password** como parte de la **HTTPS_PROXY** variable.

```
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
```

Escuchamiento de proxy web en un puerto HTTPS

Ejecute los siguientes comandos si el proxy web escucha en un puerto HTTPS.

Note

Si utilizas un certificado autofirmado para el proxy web o si estás ejecutando el proxy local en un SO que no tiene soporte nativo OpenSSL ni configuraciones predeterminadas, tendrás que configurar los certificados de proxy web tal y como se describe en el [Configuración de certificados](#) en la sección de GitHub .

Los siguientes comandos tendrán un aspecto similar al modo en que configuró el proxy web para un proxy HTTP, con la excepción de que también especificará la ruta de acceso a los archivos de certificado que instaló como se describió anteriormente.

Linux/macOS

En Linux o macOS, ejecute los siguientes comandos en el terminal para configurar el proxy local que se ejecuta en su destino para utilizar un proxy web que escucha un puerto 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
```

Si tiene que autenticarse con el proxy, debe especificar un **username** y **password** como parte de la **HTTPS_PROXY** variable.

```
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

En Windows, ejecute los comandos siguientes en el cmd para configurar e iniciar el proxy local que se ejecuta en su destino para utilizar un proxy web que escucha un puerto 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
```

Si tiene que autenticarse con el proxy, debe especificar un **username** y **password** como parte de la **HTTPS_PROXY** variable.

```
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
```

Comando y salida de ejemplo

A continuación se muestra un ejemplo de un comando que ejecuta en un SO Linux y el resultado correspondiente. En el ejemplo se muestra un proxy web que escucha en un puerto HTTP y cómo se puede configurar el proxy local para utilizar el proxy web en ambos `sourceydestinationModos`. Para poder ejecutar estos comandos, debe haber abierto un túnel y haber obtenido los tokens de acceso del cliente para el origen y el destino. También debe haber creado el proxy local y configurado su proxy web tal como se ha descrito anteriormente.

A continuación se incluye un resumen de los pasos que se han iniciado el proxy local. El proxy local:

- Identifica la URL del proxy web para que pueda utilizar la URL para conectarse al servidor proxy.
- Establece una conexión TCP con el proxy web.
- Envía un `HTTPCONNECT`solicitud al proxy web y espera el `HTTP/1.1 200` respuesta, que indica que se ha establecido la conexión.
- Actualiza el protocolo HTTPS a WebSockets para establecer una conexión de larga duración.
- Comienza a transmitir datos a través de la conexión a los extremos del dispositivo de túnel seguro.

Note

Los siguientes comandos utilizados en los ejemplos utilizan la `verbosity` para ilustrar una descripción general de los distintos pasos descritos anteriormente después de ejecutar el proxy local. Le recomendamos que utilice este indicador solo para fines de pruebas.

Ejecución de proxy local en modo fuente

Los siguientes comandos muestran cómo ejecutar el proxy local en modo de origen.

```
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 continuación se muestra un ejemplo de resultado de cómo ejecutar el proxy local en `sourceModo`.

```
...  
  
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  
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
```

Ejecución de proxy local en modo de destino

Los siguientes comandos muestran cómo ejecutar el proxy local en 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 continuación se muestra un ejemplo de resultado de cómo ejecutar el proxy local en destino 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: 06717bffffed3fd05-00001355-000b8315-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  
Setting up web socket pings for every 5000 milliseconds  
Scheduled next read:  
...  
Starting web socket read loop continue reading...
```

Flujos de datos multiplex en un túnel seguro

Puede utilizar varios flujos de datos por túnel utilizando la función de multiplexación segura de túneles. Con la multiplexación, puede solucionar problemas de dispositivos mediante múltiples conexiones o puertos (por ejemplo, un navegador web que requiere enviar varios flujos de datos HTTP y SSH). También puede reducir la carga operativa eliminando la necesidad de crear, implementar e iniciar varios proxies locales o abrir varios túneles en el mismo dispositivo.

Ejemplo de caso de uso

Puede utilizar la función de multiplexación en caso de que un dispositivo sobre el terreno requiera más de una conexión al dispositivo para solucionarlo correctamente. Por ejemplo, es posible que

tenga que conectarse a una aplicación web en el dispositivo para cambiar algunos parámetros de red, al mismo tiempo que emita comandos shell a través del terminal para verificar que el dispositivo funciona correctamente con los nuevos parámetros de red. En este escenario, es posible que tenga que conectarse al dispositivo a través de HTTP y SSH y transferir dos flujos de datos parallel para acceder simultáneamente a la aplicación web y al terminal. Con la función de multiplexación, estos dos flujos independientes se pueden transferir a través del mismo túnel al mismo tiempo.

Cómo configurar un túnel multiplexado

El siguiente procedimiento le explicará cómo configurar un túnel multiplexado para solucionar problemas de dispositivos mediante aplicaciones que requieren conexiones a varios puertos. Configurará un túnel con dos secuencias multiplexadas: una secuencia HTTP y una transmisión SSH.

1. En primer lugar, configure el dispositivo de destino con archivos de configuración. Los archivos de configuración se pueden proporcionar en el dispositivo si es poco probable que cambien las asignaciones de puertos. En el dispositivo de destino, cree un directorio de configuración denominado config en la misma carpeta en la que se ejecuta el proxy local. A continuación, cree un archivo denominado SSHSource.in en este directorio. El contenido de este archivo es:

```
HTTP1 = 5555
SSH1 = 3333
```

Note

Puede omitir este paso si prefiere especificar la asignación de puertos a través de la CLI o no necesita iniciar el proxy local en los puertos de escucha designados.

2. A continuación, configure el dispositivo de origen con archivos de configuración. En la misma carpeta en la que se ejecuta el proxy local, cree un directorio de configuración denominado config y otorgue al proxy local el permiso de lectura a este directorio. A continuación, cree un archivo denominado SSHDestination.in en este directorio. El contenido de este archivo es:

```
HTTP1 = 80
SSH1 = 22
```

Note

Puede omitir este paso si prefiere especificar la asignación de puertos a través de la CLI. En caso afirmativo, tendrá que actualizar el de túnel (p. 784) para utilizar los parámetros nuevos.

3. Abrir un túnel con el identificador de servicio HTTP1ySSH1.thingName es opcional si el dispositivo no está registrado con AWS IoT.

```
aws iotsecuretunneling open-tunnel \
--destination-config thingName=foo,services=HTTP1,SSH1
```

Después de esta llamada, se proporcionará un token de acceso de cliente de origen y destino. Tenga en cuenta la destination_client_access_token y source_client_access_token para los siguientes pasos. El resultado debería tener parecerse al siguiente:

```
{
  "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. A continuación, inicie el proxy local de destino. Estarás conectado al servicio de túnel seguro tras la entrega de tokens. Un proxy local que se ejecuta en dispositivos de destino se inicia en modo de destino. Dispone de dos opciones para lograrlo:

1. Inicie el proxy local de destino con archivos de configuración desde el paso 1.

```
./localproxy -r us-east-1 -m dst -t destination_client_access_token
```

2. Inicie el proxy local de destino con la asignación especificada a través de la CLI.

```
./localproxy -r us-east-1 -d HTTP1=80,SSH1=22 -t destination_client_access_token
```

5. Ahora, inicie el proxy local de origen. Un proxy local que se ejecuta en dispositivos de origen se inicia en modo fuente. Dispone de tres opciones para lograrlo:

1. Inicie el proxy local de origen con archivos de configuración desde el paso 2.

```
./localproxy -r us-east-1 -m src -t source_client_access_token
```

2. Inicie el proxy local de origen con la asignación especificada a través de la CLI.

```
./localproxy -r us-east-1 -s HTTP1=5555,SSH1=3333 -t source_client_access_token
```

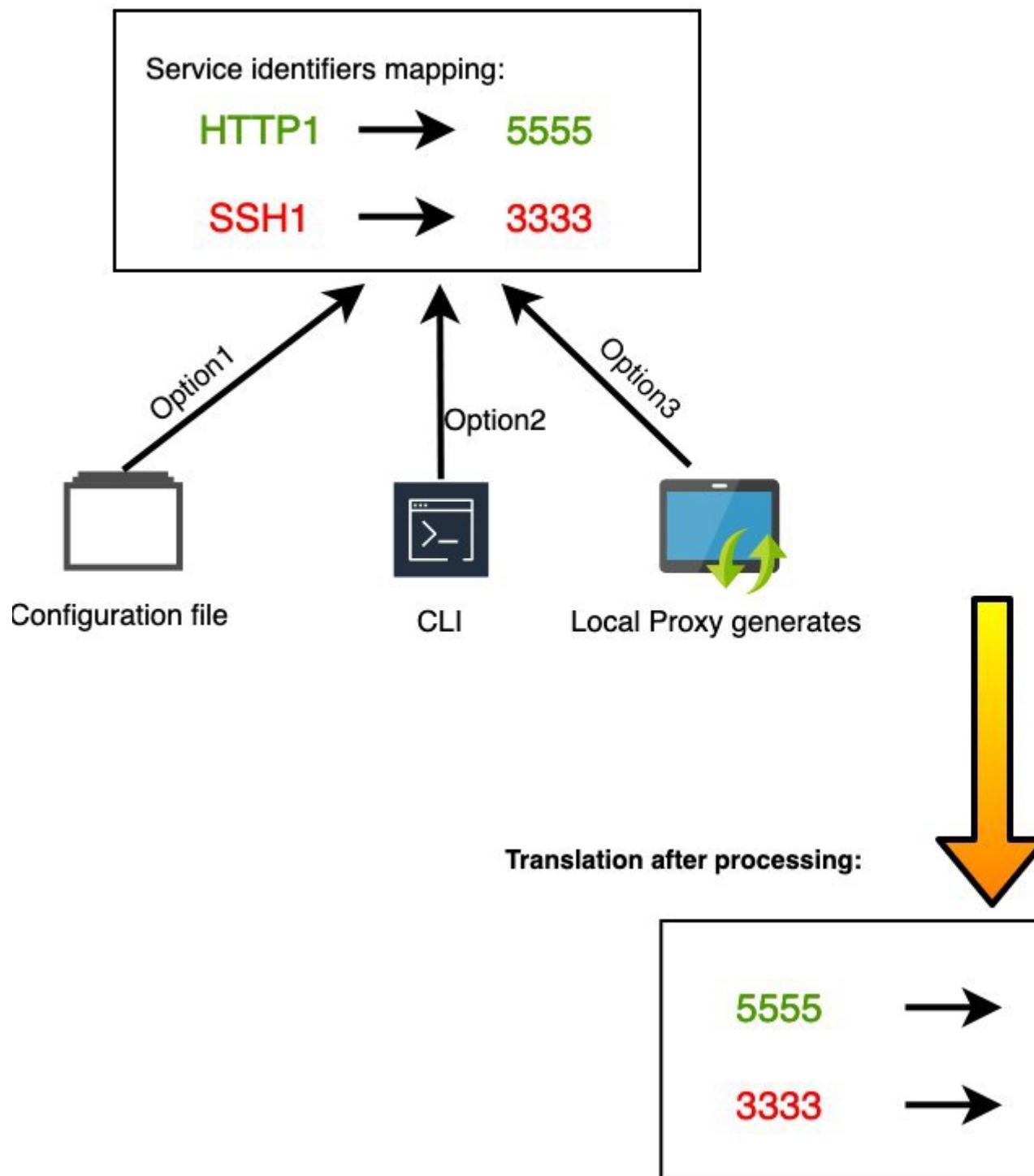
3. Inicie el proxy local de origen sin archivos de configuración ni asignación especificada desde la CLI. El proxy local recogerá los puertos disponibles para usar y administrar las asignaciones por usted.

```
./localproxy -r us-east-1 -m src -t source_client_access_token
```

6. Los datos de aplicación de la conexión SSH y HTTP ahora se pueden transferir simultáneamente a través del túnel multiplexado. Como se puede ver en el mapa siguiente, el identificador de servicio actúa como un formato legible para traducir la asignación de puertos entre el dispositivo de origen y de destino. Con esta configuración, el túnel seguro hará lo siguiente:

1. Reenvíe cualquier tráfico HTTP entrante desde el puerto 5555 del dispositivo de origen al puerto 80 del dispositivo de destino.
 2. Reenvíe cualquier tráfico SSH entrante desde el puerto 3333 del dispositivo de origen al puerto 22 del dispositivo de destino.

Source Local Proxy



Fragmento de agente de IoT

El agente IoT se utiliza para recibir el mensaje MQTT que incluye el token de acceso de cliente e iniciar un proxy local en el dispositivo remoto. Debe instalar y ejecutar el agente IoT en el dispositivo remoto si desea que la tunelización segura entregue el token de acceso de cliente. El agente de IoT debe suscribirse al siguiente tema reservado de MQTT de IoT:

```
$aws/things/thing-name/tunnels/notify
```

donde *thing-name* es el nombre del objeto de IoT asociado con el dispositivo remoto.

A continuación, se muestra un ejemplo de una carga útil de mensaje MQTT:

```
{  
    "clientAccessToken": "destination-client-access-token",  
    "clientMode": "destination",  
    "region": "aws-region",  
    "services": [destination-service]  
}
```

Después de recibir un mensaje MQTT, el agente IoT debe iniciar un proxy local en el dispositivo remoto con los parámetros apropiados.

El siguiente código Java muestra cómo utilizar el [AWS IoT SDK de dispositivos](#) y [ProcessBuilder](#) desde la biblioteca Java para crear un agente IoT sencillo que funcione con tunelización segura.

```
// Find the IoT device endpoint for your AWS account  
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);
    }
}
}
```

Control del acceso a los túneles

Tunelización segura proporciona acciones específicas de servicios, recursos y claves de contexto de condición para usarlas en las políticas de permisos de IAM.

Requisitos previos de acceso al túnel

- Describe cómo proteger AWS recursos con [Políticas de IAM](#).
- Aprenda a crear y evaluar [condiciones de IAM](#).
- Describe cómo proteger AWS recursos con [etiquetas de recursos](#).

Políticas de acceso a tú

Debe utilizar las siguientes políticas para autorizar permisos para utilizar la API de túnel seguro. Para obtener más información acerca de la seguridad de AWS IoT, consulte [Identity and Access Management en AWS IoT \(p. 384\)](#).

iot:OpenTunnel

La acción de política `iot:OpenTunnel` concede un permiso a la entidad principal para llamar a [OpenTunnel](#).

En el `navegadorResource` elemento de la declaración política de IAM:

- Especifique el ARN del túnel comodín

```
arn:aws:iot:aws-region:aws-account-id:tunnel/*
```

- Especificar un ARN de cosa para administrar el `OpenTunnel` permiso para cosas específicas de IoT:

```
arn:aws:iot:aws-region:aws-account-id:thing/thing-name
```

Por ejemplo, la siguiente instrucción de política le permite abrir un túnel con el objeto de IoT llamado `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"  
    ]  
}
```

La acción de política `iot:OpenTunnel` admite las siguientes claves de condición:

- `iot:ThingGroupArn`
- `iot:TunnelDestinationService`
- `aws:RequestTag/clave-etiqueta`
- `aws:SecureTransport`
- `aws:TagKeys`

La siguiente instrucción de política le permite abrir un túnel con el objeto si el objeto pertenece a un grupo de objetos con un nombre que comienza por `TestGroup` y el servicio de destino configurado en el túnel es 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"  
            ]  
        }  
    }  
}
```

También puede utilizar etiquetas de recursos para controlar los permisos para abrir túneles. Por ejemplo, la siguiente instrucción de política permite abrir un túnel si la clave de etiqueta `Owner` está presente con un valor de `Admin` y no se especifica ninguna otra etiqueta. Para obtener información general sobre el uso de etiquetas, consulte [Etiquetado de los recursos de 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: rotar el token de acceso a túnel

Laiot:RotateTunnelAccessToken La acción de política concede un permiso a la entidad principal para llamar a [Token de acceso de túnel de rotación](#).

En el navegador Resource elemento de la declaración política de IAM:

- Especifique un ARN de túnel completo:

```
arn:aws:iot:aws-region: aws-account-id:tunnel/tunnel-id
```

También puede utilizar el ARN del túnel comodín:

```
arn:aws:iot:aws-region:aws-account-id:tunnel/*
```

- Especificar un ARN de cosa para administrar el `RotateTunnelAccessToken` permiso para cosas específicas de IoT:

```
arn:aws:iot:aws-region:aws-account-id:thing/thing-name
```

Por ejemplo, la siguiente instrucción de política le permite rotar el token de acceso de origen de un túnel o un token de acceso de destino de cliente para el objeto IoT denominado `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"
    ]
}
```

La acción de política `iot:RotateTunnelAccessToken` admite las siguientes claves de condición:

- `iot:ThingGroupArn`
- `iot:TunnelDestinationService`
- `iot:ClientMode`
- `aws:SecureTransport`

La siguiente instrucción de política le permite rotar el token de acceso de destino a la cosa si la cosa pertenece a un grupo de objetos con un nombre que comienza por `TestGroup`, el servicio de destino configurado en el túnel es SSH y el cliente está en `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

La acción de política `iot:DescribeTunnel` concede un permiso a la entidad principal para llamar a [DescribeTunnel](#).

En el navegadorResourceelemento de la declaración de política de IAM, especifique un ARN de túnel de túnel completo:

`arn:aws:iot:aws-region: aws-account-id:tunnel/tunnel-id`

También puede utilizar el ARN del túnel comodín:

`arn:aws:iot:aws-region:aws-account-id:tunnel/*`

La acción de política `iot:DescribeTunnel` admite las siguientes claves de condición:

- `aws:ResourceTag/tag-key`
- `aws:SecureTransport`

La siguiente instrucción de política le permite llamar a `DescribeTunnel` si el túnel solicitado está etiquetado con la clave `Owner` con un 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

La acción de política `iot>ListTunnels` concede un permiso a la entidad principal para llamar a [ListTunnels](#).

En el navegadorResourceelemento de la declaración política de IAM:

- Especifique el ARN del túnel comodín

`arn:aws:iot:aws-region:aws-account-id:tunnel/*`

- Especificar un ARN de cosa para administrar el `ListTunnels` permiso para elementos de IoT seleccionados:

```
arn:aws:iot:aws-region:aws-account-id:thing/thing-name
```

La acción de política admite la clave de condición `aws:SecureTransport`.

La siguiente instrucción de política le permite mostrar los túneles del objeto denominado `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

La acción de política `iot:ListTagsForResource` concede un permiso a la entidad principal para llamar a `ListTagsForResource`.

En el navegador `Resource` elemento de la declaración de política de IAM, especifique un ARN de túnel de túnel completo:

```
arn:aws:iot:aws-region: aws-account-id:tunnel/tunnel-id
```

También puede utilizar el ARN del túnel comodín:

```
arn:aws:iot:aws-region:aws-account-id:tunnel/*
```

La acción de política `iot:ListTagsForResource` admite la clave de condición `aws:SecureTransport`.

iot:CloseTunnel

La acción de política `iot:CloseTunnel` concede un permiso a la entidad principal para llamar a `CloseTunnel`.

En el navegador `Resource` elemento de la declaración de política de IAM, especifique un ARN de túnel de túnel completo:

```
arn:aws:iot:aws-region: aws-account-id:tunnel/tunnel-id
```

También puede utilizar el ARN del túnel comodín:

```
arn:aws:iot:aws-region:aws-account-id:tunnel/*
```

La acción de política `iot:CloseTunnel` admite las siguientes claves de condición:

- `iot:Delete`
- `aws:ResourceTag/tag-key`
- `aws:SecureTransport`

La siguiente instrucción de política le permite llamar a `CloseTunnel` si el parámetro `Delete` de la solicitud es `false` y el túnel solicitado está etiquetado con la clave `Owner` y el valor `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

La acción de política `iot:TagResource` concede un permiso a la entidad principal para llamar a `TagResource`.

En el navegadorResourceelemento de la declaración de política de IAM, especifique un ARN de túnel de túnel completo:

`arn:aws:iot:aws-region: aws-account-id:tunnel/tunnel-id`

También puede utilizar el ARN del túnel comodín:

`arn:aws:iot:aws-region:aws-account-id:tunnel/*`

Laiot:TagResourceLa acción de política admite la clave de condición`aws:SecureTransport`.

iot:UntagResource

La acción de política `iot:UntagResource` concede un permiso a la entidad principal para llamar a `UntagResource`.

En el navegadorResourceelemento de la declaración de política de IAM, especifique un ARN de túnel de túnel completo:

`arn:aws:iot:aws-region: aws-account-id:tunnel/tunnel-id`

También puede utilizar el ARN del túnel comodín:

`arn:aws:iot:aws-region:aws-account-id:tunnel/*`

Laiot:UntagResourceLa acción de política admite la clave de condición`aws:SecureTransport`.

Configuración de un dispositivo remoto

Si desea entregar el token de acceso de cliente de destino al dispositivo remoto a través de otros métodos, en lugar de suscribirse al tema reservado de MQTT de IoT, puede que necesite dos componentes en el dispositivo remoto:

- Un agente de escucha de token de acceso de cliente de destino.
- Un proxy local.

El agente de escucha de token de acceso de cliente de destino debe funcionar con el mecanismo de entrega de token de acceso de cliente de su elección. Debe poder iniciar un proxy local en modo de destino.

ResolverAWS IoTproblemas de conectividad de túnel seguro mediante la rotación de tokens de acceso de clientes

Cuando se utilizaAWS IoT túnel seguro, podría tener problemas de conectividad incluso si el túnel está abierto. En las siguientes secciones se muestran algunos problemas posibles y cómo puede resolverlos girando los tokens de acceso del cliente. Para rotar el token de acceso de cliente (CAT), utilice la[Token de acceso de túnel de rotaciónAPI](#) o la[token de acceso de túnel de rotación AWS CLI](#). En función de si se produce un error al utilizar el cliente en el modo de origen o destino, puede rotar el CAT en modo de origen o destino, o ambos.

Note

- Si no está seguro de si es necesario rotar el CAT en el origen o en el destino, puede rotar el CAT tanto en el origen como en el destino configurando `clientMode` a ALL cuando se utiliza `elRotateTunnelAccessTokenAPI`.
- La rotación del CAT no extiende la duración del túnel. Por ejemplo, supongamos que la duración del túnel es de 12 horas y que el túnel ya ha estado abierto durante 4 horas. Al rotar los tokens de acceso, los nuevos tokens que se generan solo se pueden utilizar durante las 8 horas restantes.

Temas

- [Error de token de acceso de cliente no válido \(p. 791\)](#)
- [Error de discrepancia de token de cliente \(p. 791\)](#)
- [Problemas de conectividad de dispositivos \(p. 792\)](#)

Error de token de acceso de cliente no válido

Cuando se utilizaAWS IoT túnel seguro, puede encontrarse con un error de conexión al utilizar el mismo token de acceso de cliente (CAT) para volver a conectarse al mismo túnel. En este caso, el proxy local no puede conectarse al servidor proxy de túnel seguro. Si utilizas un cliente en modo de origen, puede aparecer el siguiente mensaje de error:

```
Invalid access token: The access token was previously used and cannot be used again
```

El error se produce porque el proxy local solo puede usar el token de acceso de cliente (CAT) una vez y, a continuación, no es válido. Para resolver este error, gire el token de acceso de cliente en el `SOURCE` modo para generar un nuevo CAT para el origen. Para ver un ejemplo que muestra cómo rotar el CAT de origen, consulte[Ejemplo de CAT de origen de rotación \(p. 792\)](#).

Error de discrepancia de token de cliente

Note

No se recomienda utilizar tokens de cliente para reutilizar el CAT. Le recomendamos que utilice `elRotateTunnelAccessTokenAPI` en su lugar para rotar los tokens de acceso del cliente para volver a conectarse al túnel.

Si utiliza tokens de cliente, puede reutilizar el CAT para volver a conectarse al túnel. Para reutilizar el CAT, debe proporcionar el token de cliente con el CAT la primera vez que se conecte a un túnel seguro. El túnel

seguro almacena el token de cliente, por lo que para los intentos de conexión posteriores utilizando el mismo token, también se debe proporcionar el token de cliente.

Al utilizar tokens de cliente, si utilizas un cliente en modo fuente, es posible que aparezca el siguiente error:

```
Invalid client token: The provided client token does not match the client token that was previously set.
```

El error se produce porque el token de cliente proporcionado no coincide con el token de cliente que se proporcionó con el CAT al acceder al túnel. Para solucionar este error, gire el CAT en el `SOURCE` modo para generar un nuevo CAT para el origen. A continuación se muestra un ejemplo:

Ejemplo de CAT de origen de rotación

A continuación, se muestra un ejemplo de cómo ejecutar la `RotateTunnelAccessTokenAPI` en el `SOURCE` modo para generar un nuevo CAT para el origen:

```
aws iotsecuretunneling rotate-tunnel-access-token \
--region <region> \
--tunnel-id <tunnel-id> \
--client-mode SOURCE
```

La ejecución de este comando genera un nuevo token de acceso de origen y devuelve el ARN del túnel.

```
{
  "sourceAccessToken": "<source-access-token>",
  "tunnelArn": "arn:aws:iot:<region>:<account-id>:tunnel/<tunnel-id>"}
```

Ahora puede utilizar el nuevo token de origen para conectar el proxy local en modo fuente.

```
export AWSIOT_TUNNEL_ACCESS_TOKEN=<source-access-token>
./localproxy -r <region> -s <port>
```

A continuación se muestra un resultado de ejemplo de ejecución del 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 conectividad de dispositivos

Cuando se utiliza AWS IoT túnel seguro, el dispositivo podría desconectarse inesperadamente incluso si el túnel está abierto. Para identificar si un dispositivo sigue conectado al túnel, puede utilizar el `DescribeTunnelAPI` o la `describe-túnel` AWS CLI.

Un dispositivo se puede desconectar por varios motivos. Para resolver el problema de conectividad, puede rotar el CAT en el destino si el dispositivo se desconectó por los siguientes motivos posibles:

- El CAT del destino quedó inválido.
- El token no se entregó al dispositivo a través del tema MQTT reservado de túnel seguro:

```
$aws/things/<thing-name>/tunnels/notify
```

El siguiente ejemplo muestra cómo resolver este problema:

Ejemplo CAT de rotación de destino

Considere un dispositivo remoto, <RemoteThing1>. Para abrir un túnel para esa cosa, puede utilizar el siguiente comando:

```
aws iotsecuretunneling open-tunnel \
--region <region> \
--destination-config thingName=<RemoteThing1>,services=SSH
```

Al ejecutar este comando se generan los detalles del túnel y el CAT para su origen y destino.

```
{
  "sourceAccessToken": "<source-access-token>",
  "destinationAccessToken": "<destination-access-token>",
  "tunnelId": "<tunnel-id>",
  "tunnelArn": "arn:aws:iot:<region>:<account-id>:tunnel/<tunnel-id>"
}
```

Sin embargo, cuando utilizas el [DescribeTunnelAPI](#), la salida indica que el dispositivo se ha desconectado, como se muestra a continuación:

```
aws iotsecuretunneling describe-tunnel \
--tunnel-id <tunnel-id> \
--region <region>
```

Al ejecutar este comando, se muestra que el dispositivo sigue sin estar conectado.

```
{
  "tunnel": {
    ...
    "destinationConnectionState": {
      "status": "DISCONNECTED"
    },
    ...
  }
}
```

Para solucionar este error, ejecute la [RotateTunnelAccessTokenAPI](#) con el cliente en DESTINACIÓN modo y las configuraciones del destino. La ejecución de este comando revoca el token de acceso anterior, genera un nuevo token y vuelve a enviar este token al tema 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>
```

Al ejecutar este comando se genera el nuevo token de acceso como se muestra a continuación. A continuación, el token se entrega al dispositivo para conectarse al túnel, si el agente del dispositivo está configurado correctamente.

```
{
```

```
        "destinationAccessToken": "destination-access-token",  
        "tunnelArn": "arn:aws:iot:region:account-id:tunnel/tunnel-id"  
    }
```

Aprovisionamiento de dispositivos

AWS proporciona varias formas diferentes de aprovisionar un dispositivo e instalar certificados de cliente únicos en él. En esta sección se describen todas las formas y cómo seleccionar la mejor para su solución de IoT. Estas opciones se describen en detalle en el libro blanco titulado, [Fabricación y aprovisionamiento de dispositivos con certificados X.509 en AWS IoT Core](#).

Selecciona la opción que mejor se adapte a tu situación

- Puede instalar certificados en dispositivos IoT antes de entregarlos

Si puede instalar de forma segura certificados de cliente únicos en sus dispositivos IoT antes de entregarlos para su uso por el usuario final, desea utilizar [justo a tiempo aprovisionamiento \(JITP\) \(p. 804\)](#) o [justo a tiempo registro \(JITR\) \(p. 311\)](#).

Mediante JITP y JITR, la entidad de certificación (CA) utilizada para firmar el certificado de dispositivo está registrada con AWS IoT y es reconocida por AWS IoT cuando el dispositivo se conecta por primera vez. El dispositivo se aprovisiona en AWS IoT en su primera conexión utilizando los detalles de su plantilla de aprovisionamiento.

Para obtener más información sobre una sola cosa, JITP, JITR y aprovisionamiento masivo de dispositivos que tienen certificados exclusivos, consulte [the section called “Aprovisionamiento de dispositivos que tienen certificados de dispositivo” \(p. 803\)](#).

- Los usuarios finales o los instaladores pueden utilizar una aplicación para instalar certificados en sus dispositivos IoT

Si no puede instalar de forma segura certificados de cliente únicos en su dispositivo IoT antes de entregarlos al usuario final, pero el usuario final o un instalador pueden utilizar una aplicación para registrar los dispositivos e instalar los certificados de dispositivo únicos, desea utilizar [el aprovisionamiento por usuario de confianza \(p. 799\)](#) Proceso.

El uso de un usuario de confianza, como un usuario final o un instalador con una cuenta conocida, puede simplificar el proceso de fabricación del dispositivo. En lugar de un certificado de cliente único, los dispositivos tienen un certificado temporal que permite que el dispositivo se conecte a AWS IoT durante solo 5 minutos. Durante esa ventana de 5 minutos, el usuario de confianza obtiene un certificado de cliente único con una vida útil más larga y lo instala en el dispositivo. La duración limitada del certificado de reclamación minimiza el riesgo de un certificado comprometido.

Para obtener más información, consulte [the section called “Aprovisionamiento por usuario de confianza” \(p. 799\)](#).

- Los usuarios finales NO PUEDEN usar una aplicación para instalar certificados en sus dispositivos IoT

Si ninguna de las opciones anteriores funcionará en su solución IoT, el [Aprovisionamiento por reclamación \(p. 797\)](#) process es una opción. Con este proceso, sus dispositivos IoT tienen un certificado de reclamación que comparten otros dispositivos de la flota. La primera vez que un dispositivo se conecta con un certificado de reclamación, AWS IoT registra el dispositivo mediante su plantilla de aprovisionamiento y emite al dispositivo su certificado de cliente exclusivo para obtener acceso posterior a AWS IoT.

Esta opción permite el aprovisionamiento automático de un dispositivo cuando se conecta a AWS IoT, pero podría suponer un riesgo mayor en caso de que se produzca un certificado de reclamación comprometido. Si un certificado de reclamación se ve comprometido, puede desactivarlo. Al desactivar el certificado de reclamación se impide que todos los dispositivos con ese certificado de reclamación

se registren en el futuro. Sin embargo; la desactivación del certificado de reclamación no bloquea los dispositivos que ya se han aprovisionado.

Para obtener más información, consulte [the section called “Aprovisionamiento por reclamación” \(p. 797\)](#).

Dispositivos de aprovisionamiento enAWS IoT

Cuando aprovisione un dispositivo con AWS IoT, debe crear recursos para que AWS IoT y los dispositivos puedan comunicarse de forma segura. Se pueden crear otros recursos que le ayuden a administrar su flota de dispositivos. Durante el proceso de aprovisionamiento se pueden crear los siguientes recursos:

- Un objeto de IoT.

Los objetos de IoT son entradas del registro de dispositivos de AWS IoT. Cada objeto tiene un nombre único y un conjunto de atributos, y está asociado con un dispositivo físico. Los objetos se pueden definir usando un tipo de objeto o agruparse en grupos de objetos. Para obtener más información, consulte [Administración de dispositivos con AWS IoT \(p. 266\)](#).

Aunque no es necesario, la creación de objetos permite administrar la flota de dispositivos de manera más eficaz mediante la búsqueda de dispositivos por tipo de objeto, grupo de objetos y atributos de objetos. Para obtener más información, consulte [Indexación de flotas \(p. 825\)](#).

- Un certificado X.509

Los dispositivos utilizan certificados X.509 para realizar la autenticación mutua con AWS IoT. Puede registrar un certificado existente o hacer que AWS IoT genere y registre un nuevo certificado por usted. Un certificado se asocia a un dispositivo asociándolo al objeto que representa el dispositivo. También debe copiar el certificado y la clave privada asociada en el dispositivo. Los dispositivos presentan el certificado al conectarse a AWS IoT. Para obtener más información, consulte [Autenticación \(p. 294\)](#).

- Una política de IoT

Las políticas de IoT definen qué operaciones puede realizar un dispositivo en AWS IoT. Las políticas de IoT se asocian a certificados de dispositivo. Cuando un dispositivo presenta el certificado a AWS IoT, se conceden los permisos especificados en la política. Para obtener más información, consulte [Autorización \(p. 331\)](#). Cada dispositivo necesita un certificado para comunicarse con AWS IoT.

AWS IoT permite el aprovisionamiento automatizado de flotas mediante plantillas de aprovisionamiento. Las plantillas de aprovisionamiento describen los recursos que AWS IoT necesita para aprovisionar el dispositivo. Las plantillas contienen variables que permiten utilizar una plantilla para aprovisionar varios dispositivos. Cuando aprovisione un dispositivo, especifique valores para las variables específicas de este utilizando un diccionario o mapa. Para aprovisionar otro dispositivo, especifique nuevos valores en el diccionario.

Puede utilizar el aprovisionamiento automatizado tanto si sus dispositivos tienen certificados únicos (y su clave privada asociada) como si no.

API de aprovisionamiento de flotas

Existen varias categorías de API utilizadas en el aprovisionamiento de flotas:

- Estas funciones de plano de control crean y administran las plantillas de aprovisionamiento de flotas y configuran políticas de usuario de confianza.
 - [CreateProvisioningTemplate](#)

- [CreateProvisioningTemplateVersion](#)
- [DeleteProvisioningTemplate](#)
- [DeleteProvisioningTemplateVersion](#)
- [DescribeProvisioningTemplate](#)
- [DescribeProvisioningTemplateVersion](#)
- [ListProvisioningTemplates](#)
- [ListProvisioningTemplateVersions](#)
- [UpdateProvisioningTemplate](#)
- Los usuarios de confianza pueden utilizar esta función de plazo de control para generar una notificación de incorporación temporal. Esta reclamación temporal se pasa al dispositivo durante la configuración de wifi o un método similar.
 - [CreateProvisioningClaim](#).
- La API MQTT utilizada durante el proceso de aprovisionamiento por los dispositivos con un certificado de notificación de aprovisionamiento incrustado en un dispositivo o pasado a él por un usuario de confianza.
 - [the section called “CreateCertificateFromCsr” \(p. 819\)](#)
 - [the section called “CreateKeysAndCertificate” \(p. 821\)](#)
 - [the section called “RegisterThing” \(p. 823\)](#)

Aprovisionamiento de dispositivos que no tienen certificados de dispositivo mediante el aprovisionamiento de flotas

Mediante el uso de AWS IoT provisionamiento de flotas, AWS IoT puede generar y distribuir de forma segura certificados de dispositivo y claves privadas a sus dispositivos cuando estos se conectan a AWS IoT por primera vez. AWS IoT proporciona certificados de cliente firmados por la entidad de certificación (CA) Amazon Root.

Existen dos formas de utilizar el aprovisionamiento de flotas:

- Por reclamación
- Por usuario de confianza

Aprovisionamiento por reclamación

Los dispositivos se pueden fabricar con un certificado de notificación de aprovisionamiento y una clave privada (que son credenciales de uso especial) incrustados. Si estos certificados están registrados con AWS IoT, el servicio puede intercambiarlos por certificados de dispositivo únicos que el dispositivo puede usar para realizar las operaciones normales. El proceso consta de los pasos siguientes:

Antes de entregar el dispositivo

1. Llamada [CreateProvisioningTemplate](#) para crear una plantilla de aprovisionamiento. Esta API devuelve un ARN de plantilla. Para obtener más información, consulte [API de MQTT de aprovisionamiento de dispositivos \(p. 819\)](#).

También puede crear una plantilla de aprovisionamiento de flotas desde la consola de AWS IoT.

- a. En el panel de navegación, seleccione [Conectary](#), a continuación, seleccione [Plantillas de aprovisionamiento de](#).

- b. ElegirCrear plantillay siga las instrucciones.
2. Cree los certificados y las claves privadas asociadas que se utilizarán como certificados de reclamación de aprovisionamiento.
 3. Registre estos certificados en AWS IoT y asocie una política de IoT que restrinja el uso de los certificados. El siguiente ejemplo de política de IoT restringe el uso del certificado asociado a esta política para aprovisionar dispositivos.

```
{  
    "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. Cuando aprovisiona dispositivos, puede conceder al servicio de AWS IoT permiso para crear o actualizar recursos de IoT, como objetos y certificados de la cuenta. Para ello, asocie la política administrada AWSIoTThingsRegistration a un rol de IAM (denominado rol de aprovisionamiento) que confíe en la entidad principal del servicio de AWS IoT.
5. Fabricar el dispositivo con el certificado de reclamación de aprovisionamiento incrustado de forma segura en él.

El dispositivo ya está listo para ser entregado a donde se instalará para su uso.

Important

Las claves privadas de la notificación de aprovisionamiento deben estar protegidas en todo momento, también en el dispositivo. Recomendamos utilizarAWS IoT Métricas y registros de CloudWatch para comprobar si hay indicios de un uso incorrecto. Si detecta un uso incorrecto, deshabilite el certificado de notificación de aprovisionamiento para que no se pueda utilizar para el aprovisionamiento de dispositivos.

Para inicializar el dispositivo para su uso

1. El dispositivo utiliza el [AWS IoT SDK de dispositivos, SDK móviles y AWS IoT Client de dispositivos \(p. 1274\)](#) para conectarse y autenticarse con AWS IoT utilizando el certificado de reclamación de aprovisionamiento que está instalado en el dispositivo.

Note

Por seguridad, el `certificateOwnershipToken` devuelto por [CreateCertificateFromCsr](#) (p. 819) y [CreateKeysAndCertificate](#) (p. 821) caduca después de una hora. [RegisterThing](#) (p. 823) debe llamarse antes de que `certificateOwnershipTokenExpires`. Si el certificado creado por [CreateCertificateFromCsr](#) (p. 819) o [CreateKeysAndCertificate](#) (p. 821) no se ha activado y no se ha adjuntado a una política o a una cosa en el momento en que caduca el token, se elimina el certificado. Si el token caduca, el dispositivo puede llamar [CreateCertificateFromCsr](#) (p. 819) o [CreateKeysAndCertificate](#) (p. 821) de nuevo para generar un nuevo certificado.

2. El dispositivo obtiene un certificado permanente y una clave privada mediante una de estas opciones. El dispositivo utilizará el certificado y la clave para toda la autenticación futura con AWS IoT.
 - a. Llamada [CreateKeysAndCertificate](#) (p. 821) para crear un nuevo certificado y una clave privada mediante la AWSentidad de certificación.

O bien

 - b. Llamada [CreateCertificateFromCsr](#) (p. 819) para generar un certificado desde una solicitud de firma de certificado que mantiene su clave privada segura.
3. Desde el dispositivo, llame a [RegisterThing](#) (p. 823) para registrar el dispositivo con AWS IoT y crear recursos en la nube.

El servicio de aprovisionamiento de flotas crea recursos en la nube como objetos, grupos de objetos y atributos, tal como se definen en la plantilla de aprovisionamiento.

4. Despues de guardar el certificado permanente en el dispositivo, este debe desconectarse de la sesión que inició con el certificado de reclamación de aprovisionamiento y volver a conectarse con el certificado permanente.

El dispositivo ya está listo para comunicarse normalmente con AWS IoT.

Aprovisionamiento por usuario de confianza

En muchos casos, un dispositivo se conecta a AWS IoT por primera vez cuando un usuario de confianza, como un usuario final o un técnico de la instalación, utiliza una aplicación móvil para configurar el dispositivo en su ubicación implementada.

Important

Debe administrar el acceso y el permiso del usuario de confianza para realizar este procedimiento. Una forma de hacerlo es proporcionar y mantener una cuenta para el usuario de confianza que lo autentica y le otorga acceso a las características y las API de AWS IoT necesarias para realizar este procedimiento.

Antes de entregar el dispositivo

1. Llamada [CreateProvisioningTemplate](#) para crear una plantilla de aprovisionamiento y devolver su `templateArn` y `templateName`.
2. Cree un rol de IAM que utilice un usuario de confianza para iniciar el proceso de aprovisionamiento. La plantilla de aprovisionamiento solo permite a ese usuario aprovisionar un dispositivo. Por ejemplo:

```
{  
    "Effect": "Allow",  
    "Action": [  
        "iot:CreateProvisioningClaim"  
    ]  
}
```

```
        ],
    "Resource": [
        "arn:aws:aws-region:aws-account-id:provisioningtemplate/templateName"
    ]
}
```

3. Cuando aprovisiona dispositivos, puede conceder al servicio de AWS IoT permiso para crear o actualizar recursos de IoT, como objetos y certificados de la cuenta. Para ello, asocie elAWSIoTThingsRegistrationAdministrar la política de un rol de IAM (denominadrol de aprovisionamiento) que confía en elAWS IoTPrincipal del servicio.
4. Proporcione los medios para identificar a sus usuarios de confianza, por ejemplo proporcionándoles una cuenta que pueda autenticarlos y autorizar sus interacciones con elAWSAPI necesarias para registrar sus dispositivos.

Para inicializar el dispositivo para su uso

1. El usuario de confianza inicia sesión en su aplicación móvil o servicio web de aprovisionamiento.
2. La aplicación móvil o la aplicación web utiliza el rol de IAM y llama aCreateProvisioningClaimpara obtener un certificado de reclamación de aprovisionamiento temporal deAWS IoT.

Note

Por motivos de seguridad, el certificado de la reclamación de aprovisionamiento temporal queCreateProvisioningClaimLas devoluciones caducan después de cinco minutos. Los pasos siguientes deben devolver correctamente un certificado válido antes de que caduque el certificado de reclamación de aprovisionamiento temporal. Los certificados de reclamación de aprovisionamiento temporal no aparecen en la lista de certificados de su cuenta.

3. La aplicación móvil o la aplicación web proporciona el certificado de reclamación de aprovisionamiento temporal al dispositivo junto con cualquier información de configuración necesaria, como credenciales Wi-Fi.
4. El dispositivo utiliza el certificado de reclamación de aprovisionamiento temporal para conectarse a AWS IoT mediante el [AWS IoTSDK de dispositivos, SDK móviles yAWS IoTCliente de dispositivos \(p. 1274\)](#).
5. El dispositivo obtiene un certificado permanente y una clave privada mediante una de estas opciones en un plazo de cinco minutos desde la conexión aAWS IoTcon el certificado de la reclamación de aprovisionamiento temporal. El dispositivo utilizará el certificado y clave de que estas opciones devuelvan para toda la autenticación futura conAWS IoT.
 - a. Llamada[CreateKeysAndCertificate \(p. 821\)](#)para crear un nuevo certificado y una clave privada mediante laAWSentidad de certificación.
O bien
 - b. Llamada[CreateCertificateFromCsr \(p. 819\)](#)para generar un certificado desde una solicitud de firma de certificado que mantiene su clave privada segura.

Note

Recuerde[CreateKeysAndCertificate \(p. 821\)](#)o[CreateCertificateFromCsr \(p. 819\)](#)debe devolver un certificado válido en un plazo de cinco minutos desde la conexión aAWS IoTcon el certificado de la reclamación de aprovisionamiento temporal.

6. El dispositivo llama a [RegisterThing \(p. 823\)](#) para registrar el dispositivo con AWS IoT y crear recursos en la nube.

El servicio de aprovisionamiento de flotas crea recursos en la nube como objetos de IoT, grupos de objetos y atributos, tal como se definen en la plantilla de aprovisionamiento.

- Después de guardar el certificado permanente en el dispositivo, el dispositivo debe desconectarse de la sesión que inició con el certificado de reclamación de aprovisionamiento temporal y volver a conectarse con el certificado permanente.

El dispositivo ya está listo para comunicarse normalmente con AWS IoT.

Uso de enlaces de preaprovisionamiento con la CLI de AWS

El siguiente procedimiento crea una plantilla de aprovisionamiento con enlaces de preaprovisionamiento. La función Lambda utilizada aquí es un ejemplo que se puede modificar.

Para crear y aplicar un enlace de preaprovisionamiento a una plantilla de aprovisionamiento

- Cree una función Lambda que tenga una entrada y salida definidas. Las funciones de Lambda son muy personalizables `allowProvisioning` y `parameterOverrides` son necesarios para crear enlaces de preaprovisionamiento. Para obtener más información acerca de la creación de funciones Lambda, consulte [Uso de AWS Lambda con AWS Command Line Interface](#).

El siguiente es un ejemplo de una salida de función de Lambda:

```
{  
    "allowProvisioning": True,  
    "parameterOverrides": {  
        "incomingKey0": "incomingValue0",  
        "incomingKey1": "incomingValue1"  
    }  
}
```

- AWS IoT utiliza políticas basadas en recursos para llamar a Lambda, por lo que debe dar AWS IoT permiso para llamar a la función Lambda.

Important

Asegúrese de incluir el `source-arn` en las claves de contexto de condición global de las políticas asociadas a la acción de Lambda para evitar la manipulación de permisos. Para obtener más información acerca de este tema, consulte [Prevención del suplente confuso entre servicios \(p. 343\)](#).

A continuación se muestra un ejemplo de cómo se utiliza `permiso de complemento` da permiso de IoT a tu Lambda.

```
aws lambda add-permission \  
    --function-name myLambdaFunction \  
    --statement-id iot-permission \  
    --action lambda:InvokeFunction \  
    --principal iot.amazonaws.com
```

- Agregue un enlace de preaprovisionamiento a una plantilla mediante el comando `create-provisioning-template` o `update-provisioning-template`.

En el siguiente ejemplo de CLI se utiliza la `create-provisioning-template` para crear una plantilla de aprovisionamiento que tenga enlaces de preaprovisionamiento:

```
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 \  
    --region us-east-1
```

```
--pre-provisioning-hook file://hooks.json
```

El resultado de este comando tendrá un aspecto similar al siguiente.

```
{  
    "templateArn": "arn:aws:iot:us-east-1:1234564789012:provisioningtemplate/  
myTemplate",  
    "defaultVersionId": 1,  
    "templateName": myTemplate  
}
```

También puede cargar un parámetro desde un archivo en lugar de escribirlo todo como un valor de parámetro de línea de comandos para ahorrar tiempo. Para obtener más información, consulte [Cargando AWS CLI Parámetros de un archivo](#). A continuación se muestra el parámetro template en 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:Connect",  
                            "Subject": "deviceArn",  
                            "Resource": "*"  
                        }  
                    ]  
                }  
            }  
        }  
    }  
}
```

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

A continuación se muestra el parámetro `pre-provisioning-hook` en formato JSON expandido:

```
{
    "targetArn" : "arn:aws:lambda:us-east-1:765219403047:function:pre_provisioning_test",
    "payloadVersion" : "2020-04-01"
}
```

Aprovisionamiento de dispositivos que tienen certificados de dispositivo

AWS IoT proporciona tres formas de aprovisionar dispositivos cuando ya tienen un certificado de dispositivo (y una clave privada asociada):

- Aprovisionamiento de un solo objeto con una plantilla de aprovisionamiento. Esta es una buena opción si solo necesita aprovisionar los dispositivos de uno en uno.
- Aprovisionamiento "just-in-time" (JITP) con una plantilla que aprovisiona un dispositivo cuando se conecta por primera vez a AWS IoT. Esta es una buena opción si necesita registrar un gran número de dispositivos, pero no dispone de información sobre ellos que se pueda incluir en una lista de aprovisionamiento por lotes.
- Registro masivo Esta opción le permite especificar una lista de valores de aprovisionamiento de un solo objeto que se almacenan en un archivo en un bucket de S3. Este enfoque funciona bien si tiene un gran número de dispositivos conocidos cuyas características puede incluir en una lista.

Temas

- [Aprovisionamiento de un solo objeto \(p. 803\)](#)
- [Aprovisionamiento justo a tiempo \(p. 804\)](#)
- [Registro masivo \(p. 807\)](#)

Aprovisionamiento de un solo objeto

Para aprovisionar un objeto, use la API de `RegisterThing` o el comando de la CLI `register-thing`. El comando de la CLI `register-thing` adopta los argumentos siguientes:

--template-body

La plantilla de aprovisionamiento.

--parameters

Una lista de pares de nombre-valor para los parámetros utilizados en la plantilla de aprovisionamiento, en formato JSON (por ejemplo, { "ThingName" : "MyProvisionedThing", "CSR" : "csr-text" }).

Consulte [Aprovisionamiento de plantillas \(p. 808\)](#).

[RegisterThing](#) o `register-thing` devuelve los ARN para los recursos y el texto del certificado que ha creado:

```
{  
    "certificatePem": "certificate-text",  
    "resourceArns": {  
        "PolicyLogicalName": "arn:aws:iot:us-west-2:123456789012:policy/2A6577675B7CD1823E271C7AAD8184F44630FFD7",  
        "certificate": "arn:aws:iot:us-west-2:123456789012:cert/cd82bb924d4c6ccbb14986dcba4f40f30d892cc6b3ce7ad5008ed6542eea2b049",  
        "thing": "arn:aws:iot:us-west-2:123456789012:thing/MyProvisionedThing"  
    }  
}
```

Si se omite el parámetro en el diccionario, se utilizará el valor predeterminado. Si no se especifica el valor predeterminado, el parámetro no se reemplazará por un valor.

Aprovisionamiento justo a tiempo

Puede tener sus dispositivos aprovisionados cuando intentan conectarse a por primera vez a AWS IoT con aprovisionamiento justo a tiempo (JITP). Para aprovisionar el dispositivo, debe habilitar el registro automático y asociar una plantilla de aprovisionamiento al certificado de CA utilizado para firmar el certificado de dispositivo. Los errores y los éxitos de aprovisionamiento se registran como [Métricas de aprovisionamiento de dispositivos \(p. 445\)](#) en Amazon CloudWatch.

Puede realizar estas configuraciones cuando registre un nuevo certificado de CA con la API [RegisterCACertificate](#) o el comando de la CLI `register-ca-certificate`:

```
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 obtener más información, consulte [Registro de su certificado de CA](#).

También puede utilizar la API [UpdateCACertificate](#) o el comando de la CLI `update-ca-certificate` para actualizar la configuración para un certificado de CA:

```
aws iot update-ca-certificate --certificate-id caCertificateId --new-auto-registration-status ENABLE --registration-config file://your-template
```

Note

El aprovisionamiento justo a tiempo JITP llama a otros AWS IoT API de plano de control durante el proceso de aprovisionamiento. Estas llamadas podrían exceder las [AWS IoT Cuotas de limitación de configurados](#) para su cuenta y provocar una limitación controlada de las llamadas.

Contacto [AWS Atención al Soporte](#) para aumentar las cuotas de limitación controlada, si es necesario.

Cuando un dispositivo intenta conectarse a AWS IoT mediante un certificado firmado por un certificado de CA registrado, AWS IoT carga la plantilla desde el certificado de CA y la utiliza para llamar a [RegisterThing \(p. 823\)](#). El flujo de trabajo de JITP registra primero un certificado con un valor de estado de PENDING_ACTIVATION. Cuando se completa el flujo de aprovisionamiento del dispositivo, el estado del certificado cambia a ACTIVE.

AWS IoT define los siguientes parámetros que puede declarar y a los que puede hacer referencia en las plantillas de aprovisionamiento:

- `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`

Los valores de estos parámetros de plantilla de aprovisionamiento se limitan a lo que JITP puede extraer del campo de asunto del certificado del dispositivo que se va a aprovisionar. El certificado debe contener valores para todos los parámetros en el cuerpo de la plantilla. El parámetro `AWS::IoT::Certificate::Id` se refiere a un ID generado internamente, no un ID que se encuentra en el certificado. Puede obtener el valor de este ID utilizando la función `principal()` dentro de una regla de AWS IoT.

Note

Puede aprovisionar dispositivos mediante AWS IoT Core Función de aprovisionamiento «just-in-time» (JITP) sin tener que enviar toda la cadena de confianza en la primera conexión de los dispositivos a AWS IoT Core. La presentación del certificado de CA es opcional, pero el dispositivo debe enviar el [Indicación de nombre de servidor \(SNI\)](#) extensión cuando se conectan.

El siguiente archivo JSON es un ejemplo de una plantilla de JITP completa. El valor del campo `templateBody` debe ser un objeto JSON especificado como una cadena de escape y solo puede utilizar los valores de la lista anterior. Puede utilizar distintas herramientas para crear la salida JSON necesaria, como `json.dumps` (Python) o `JSON.stringify` (Node). El valor del campo `roleARN` debe ser el ARN de un rol que tenga `AWSIoTThingsRegistration` asociado. Además, la plantilla puede utilizar un `PolicyName` existente en lugar del `PolicyDocument` insertado en el ejemplo. (En el primer ejemplo se han añadido saltos de línea para facilitar su lectura, pero puede copiar y pegar la plantilla que aparece justo debajo).

```
{  
    "templateBody" : "{  
        \"Parameters\": {  
            \"AWS::IoT::Certificate::CommonName\": {  
                \"Type\": \"String\"  
            },  
            \"AWS::IoT::Certificate::SerialNumber\": {  
                \"Type\": \"String\"  
            },  
            \"AWS::IoT::Certificate::Country\": {  
                \"Type\": \"String\"  
            },  
            \"AWS::IoT::Certificate::Id\": {  
                \"Type\": \"String\"  
            },  
            \"Resources\": {  
                \"thing\": {}  
            }  
        }  
    }  
}
```

```

        \\"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
            \\\"ThingType\\\": \\\"lightBulb-versionA\\\",\\r\\n
            \\\"ThingGroups\\\": [\\r\\n
                \\\"v1-lightbulbs\\\",\\r\\n            {\\r\\n
                \\\"Ref\\\": \\\"AWS::IoT::Certificate::Country\\\"\\r\\n
        \\n            }\\r\\n        ]\\r\\n    },\\r\\n
            \\\"OverrideSettings\\\": {\\r\\n
                \\\"AttributePayload\\\": \\\"MERGE\\\",\\r\\n
                \\\"ThingType\\\": \\\"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
            \"roleArn\" : \"arn:aws:iam::123456789012:role/Provisioning-JITP"
        }
    
```

Esta es una versión que puede copiar y pegar:

```

{
    \"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
            \\\"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
                        \\\"version\\\": \"v1\",\\r\\n
                        \\\"AttributePayload\\\": {\\r\\n
                            \\\"serialNumber\\\": {\\r\\n
                                \\\"Ref\\\": \\\"AWS::IoT::Certificate::SerialNumber
        \\\"\\r\\n                            },\\r\\n
                            \\\"ThingType\\\": \\\"lightBulb-versionA\\\",\\r\\n
                            \\\"ThingGroups\\\": [\\r\\n
                                \\\"v1-lightbulbs\\\",\\r\\n                            {\\r\\n
                                \\\"Ref\\\": \\\"AWS::IoT::Certificate::Country\\\"\\r\\n
        \\n                            }\\r\\n                        ],\\r\\n
                            \\\"OverrideSettings\\\": {\\r\\n
                                \\\"AttributePayload\\\": \\\"MERGE\\\",\\r\\n
                                \\\"ThingType\\\": \\\"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          "OverrideSettings": {\r\n            "Status": "ACTIVE",\r\n            "Status": "DO_NOTHING",\r\n            "Type": "AWS::IoT::Policy",\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            "Properties": {\r\n              "RoleArn": "arn:aws:iam::123456789012:role/JITPRole"\r\n            }\r\n          }\r\n        }\r\n      }\r\n    }\r\n  }\r\n}
```

Esta plantilla de ejemplo declara valores para los parámetros de aprovisionamiento `AWS::IoT::Certificate::CommonName`, `AWS::IoT::Certificate::SerialNumber`, `AWS::IoT::Certificate::Country` y `AWS::IoT::Certificate::Id` que se extraen del certificado y se usan en la sección Resources. El flujo de trabajo de JITP utiliza después esta plantilla para llevar a cabo las siguientes acciones:

- Registrar un certificado y establecer su estado en PENDING_ACTIVE
- Crear un recurso de objeto
- Crear un recurso de política
- Asociar la política al certificado
- Asociar el certificado al objeto
- Actualizar el estado del certificado a ACTIVE

Tenga en cuenta que el aprovisionamiento del dispositivo falla si el certificado no tiene todas las propiedades mencionadas en el `Parameters` Sección sobre de `ltemplateBody`. Por ejemplo, si `AWS::IoT::Certificate::Country` se incluye en la plantilla, pero el certificado no tiene `Country`, se produce un error en el aprovisionamiento del dispositivo.

También puede usar CloudTrail para solucionar los problemas con su plantilla JITP. Para obtener información sobre las métricas registradas en Amazon CloudWatch, consulte [Métricas de aprovisionamiento de dispositivos \(p. 445\)](#).

Registro masivo

Puede utilizar el `start-thing-registration-task` para registrar cosas a la vez. Este comando toma una plantilla de registro, un nombre de bucket de S3, un nombre de clave y un ARN de rol que permite el acceso al archivo en el bucket de S3. El archivo en el bucket de S3 contiene los valores utilizados para reemplazar los parámetros de la plantilla. El archivo debe ser un archivo JSON delimitado por nueva línea. Cada línea contiene todos los valores de los parámetros para el registro de un único dispositivo. Por ejemplo:

```
{"ThingName": "foo", "SerialNumber": "123", "CSR": "csr1"}\n{"ThingName": "bar", "SerialNumber": "456", "CSR": "csr2"}
```

Las siguientes API relacionadas con el registro masivo pueden ser útiles:

- [ListThingRegistrationTasks](#): enumera las tareas de aprovisionamiento por lotes actuales.
- [DescribeThingRegistrationTask](#): Proporciona información acerca de una tarea de registro masivo de objetos específica.
- [StopThingRegistrationTask](#): Detiene una tarea de registro de objetos por lotes.
- [ListThingRegistrationTaskReports](#): Se utiliza para comprobar los resultados y errores de una tarea de registro masivo de objetos.

Note

- Solo puede ejecutarse una tarea de operación de registro masivo a la vez (por cuenta).
- Las operaciones de registro masivo llaman a otras API del plano de control de AWS IoT. Estas llamadas podrían exceder el [AWS IoT Cuotas de limitación de en la cuenta](#) y provocar errores de limitación. Contacto [AWS Atención al Support](#) para elevar tu AWS IoT Cuotas de limitación controlada, si es necesario.

Aprovisionamiento de plantillas

Una plantilla de aprovisionamiento es un documento JSON que utiliza parámetros para describir los recursos que el dispositivo debe usar para interactuar con AWS IoT. Una plantilla contiene dos secciones: `Parameters` y `Resources`. Hay dos tipos de plantillas de aprovisionamiento en AWS IoT. Uno se utiliza para el aprovisionamiento just-in-time (JITP) y el registro masivo, y otro para el aprovisionamiento de flotas.

Sección de parámetros

La sección `Parameters` declara los parámetros utilizados en la sección `Resources`. Cada parámetro declara un nombre, un tipo y un valor predeterminado opcional. El valor predeterminado se usa cuando el diccionario trasladado con la plantilla no contiene un valor para el parámetro. La sección `Parameters` de un documento de plantilla tiene el siguiente aspecto:

```
{  
    "Parameters" : {  
        "ThingName" : {  
            "Type" : "String"  
        },  
        "SerialNumber" : {  
            "Type" : "String"  
        },  
        "Location" : {  
            "Type" : "String",  
            "Default" : "WA"  
        },  
        "CSR" : {  
            "Type" : "String"  
        }  
    }  
}
```

Este snippet de plantilla declara cuatro parámetros: `ThingName`, `SerialNumber`, `Location` y `CSR`. Todos esos parámetros son de tipo `String`. El parámetro `Location` declara un valor predeterminado de `"WA"`.

Sección de recursos

La sección `Resources` de la plantilla declara los recursos necesarios para que su dispositivo se comunique con AWS IoT: un objeto, un certificado y una o más políticas de IoT. Cada recurso especifica un nombre lógico, un tipo y un conjunto de propiedades.

Un nombre lógico le permite hacer referencia a un recurso en cualquier lugar de la plantilla.

El tipo especifica la clase de recurso que está declarando. Los tipos válidos son:

- `AWS::IoT::Thing`

- `AWS::IoT::Certificate`
- `AWS::IoT::Policy`

Las propiedades que especifique dependen del tipo de recurso que está declarando.

Recursos de objetos

Los recursos de objetos se declaran mediante las siguientes propiedades:

- `ThingName`: Cadena.
- `AttributePayload`: opcional. Una lista de pares de nombre-valor.
- `ThingTypeName`: opcional. Cadena para un tipo de objeto asociado para el objeto.
- `ThingGroups`: opcional. Una lista de grupos a los que pertenece el objeto.

Recursos de certificados

Puede especificar certificados de una de las siguientes maneras:

- Una solicitud de firma de certificado (CSR).
- Un ID de certificado de un certificado de dispositivo existente. (Los ID de certificado solo se pueden utilizar con una plantilla de aprovisionamiento de flotas).
- Un certificado de dispositivo creado con un certificado de CA registrado con AWS IoT. Si tiene más de un certificado de CA registrado con el mismo campo de asunto, también debe trasladar el certificado de CA utilizado para firmar el certificado de dispositivo.

Note

Cuando declare un certificado en una plantilla, use solo uno de estos métodos. Por ejemplo, si utiliza un CSR, no puede especificar también un ID de certificado o un certificado de dispositivo. Para obtener más información, consulte [Certificados de cliente X.509 \(p. 298\)](#).

Para obtener más información, consulte [Información general del certificado X.509 \(p. 295\)](#).

Los recursos de certificados se declaran mediante las siguientes propiedades:

- `CertificateSigningRequest`: Cadena.
- `CertificateID`: Cadena.
- `CertificatePem`: Cadena.
- `CACertificatePem`: Cadena.
- `Status`: opcional. Cadena, que puede ser `ACTIVE` o `INACTIVE`. El valor predeterminado es `ACTIVE`.

Ejemplos:

- Certificado especificado con un CSR:

```
{  
    "certificate" : {  
        "Type" : "AWS::IoT::Certificate",  
        "Properties" : {  
            "CertificateSigningRequest": {"Ref" : "CSR"},  
            "Status" : "ACTIVE"  
        }  
    }  
}
```

```
    }
```

- Certificado especificado con un ID de certificado existente:

```
{  
    "certificate" : {  
        "Type" : "AWS::IoT::Certificate",  
        "Properties" : {  
            "CertificateId": {"Ref" : "CertificateId"}  
        }  
    }  
}
```

- Certificado especificado con un archivo .pem de certificado existente y un archivo .pem de certificado de CA:

```
{  
    "certificate" : {  
        "Type" : "AWS::IoT::Certificate",  
        "Properties" : {  
            "CACertificatePem": {"Ref" : "CACertificatePem"},  
            "CertificatePem": {"Ref" : "CertificatePem"}  
        }  
    }  
}
```

Recursos de políticas

Los recursos de políticas se declaran mediante una de las siguientes propiedades:

- **PolicyName**: opcional. Cadena. Hash es el valor predeterminado del documento de políticas. La `PolicyNames` solo puede hacer referencia a las políticas de IAM. Si utiliza una política de AWS IoT existente, escriba el nombre de la política para la propiedad `PolicyName`. No incluya la propiedad `PolicyDocument`.
- **PolicyDocument**: opcional. Un objeto JSON especificado como una cadena de escape. Si `PolicyDocument` no se proporciona, la política debe haberse creado ya.

Note

Si una sección `Policy` está presente, `PolicyName` o `PolicyDocument`, pero no ambas, debe especificarse.

Configuración de invalidación

Si una plantilla especifica un recurso que ya existe, la sección `OverrideSettings` le permite especificar la acción que se debe realizar:

DO NOTHING

Dejar el recurso como está.

REPLACE

Sustituir el recurso por el recurso especificado en la plantilla.

FAIL

Provocar un error en la solicitud con `ResourceConflictException`.

MERGE

Solo es válida para las propiedades `ThingGroups` y `AttributePayload` de un objeto (`thing`). Combinar los atributos o las pertenencias a grupos existentes del objeto con los especificados en la plantilla.

Cuando declara un recurso de objeto, puede especificar `OverrideSettings` para las siguientes propiedades:

- `ATTRIBUTE_PAYLOAD`
- `THING_TYPE_NAME`
- `THING_GROUPS`

Cuando declara un recurso de certificado, puede especificar `OverrideSettings` para la propiedad `Status`.

`OverrideSettings` no están disponibles para recursos de políticas.

Ejemplo de recursos

El siguiente fragmento de plantilla declara un objeto, un certificado y una 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\"] }] }"  
            }  
        }  
    }  
}
```

El objeto se declara con:

- El nombre lógico "thing".

- El tipo `AWS::IoT::Thing`.
- Un conjunto de propiedades de objeto.

Las propiedades de objeto incluyen el nombre de objeto, un conjunto de atributos, un nombre de tipo de objeto opcional y una lista opcional de grupos de objetos a los que pertenece el objeto.

Se hace referencia a los parámetros mediante `{"Ref": "parameter-name"}`. Cuando se evalúa la plantilla, los parámetros se reemplazan por el valor de los parámetros desde el diccionario trasladado con la plantilla.

El certificado se declara con:

- El nombre lógico "certificate".
- El tipo `AWS::IoT::Certificate`.
- Un conjunto de propiedades.

Las propiedades incluyen el CSR para el certificado y el establecimiento del estado en `ACTIVE`. El texto CSR se transfiere como parámetro en el diccionario transferido con la plantilla.

La política se declara con:

- El nombre lógico "policy".
- El tipo `AWS::IoT::Policy`.
- O el nombre de una política existente o un documento de políticas.

Ejemplo de plantilla para el registro JITP y masivo

El siguiente archivo JSON es un ejemplo de una plantilla de aprovisionamiento completa que especifica el certificado con un CSR:

(El valor del campo `PolicyDocument` debe ser un objeto JSON especificado como una cadena 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",  
                "ThingTypeArn" : {"Fn::GetAtt" : "thing", "ThingTypeArn"}  
            }  
        }  
    }  
}
```

```
        "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\"] } ] }"
    }
}
}
```

El siguiente archivo JSON es un ejemplo de una plantilla de aprovisionamiento completo que especifica un certificado existente con un 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\"] }] }"
            }
        }
    }
}
```

```
    }  
}
```

Aprovisionamiento de flotas

Las plantillas de aprovisionamiento de flotas las utiliza AWS IoT para definir la configuración de la nube y del dispositivo. Estas plantillas utilizan los mismos parámetros y recursos que las plantillas de registro masivo y JITP. Para obtener más información, consulte [Aprovisionamiento de plantillas \(p. 808\)](#).

Las plantillas de aprovisionamiento de flotas pueden contener una sección `Mapping` y una sección `DeviceConfiguration`. Puede utilizar funciones intrínsecas dentro de una plantilla de aprovisionamiento de flotas para generar una configuración específica del dispositivo. Las plantillas de aprovisionamiento de flotas son recursos con nombre y se identifican mediante el ARN (por ejemplo, `arn:aws:iot:us-west-2:1234568788:provisioningtemplate/templateName`).

Mapeos

La sección `Mappings` opcional hace coincidir una clave con el conjunto correspondiente de valores identificados. Por ejemplo, si desea establecer valores basándose en unAWSRegión, puede crear una asignación que utilice elRegión de AWSname como clave y contiene los valores que desea especificar para cada región específica. Puede utilizar la función intrínseca `Fn::FindInMap` para recuperar valores en una asignación.

No puede incluir parámetros, pseudoparámetros ni funciones de llamadas intrínsecas en la sección `Mappings`.

Configuración del dispositivo

La sección de configuración del dispositivo contiene los datos arbitrarios que desea enviar a sus dispositivos al realizar el aprovisionamiento. Por ejemplo:

```
{  
  "DeviceConfiguration": {  
    "Foo": "Bar"  
  }  
}
```

Si envía mensajes a sus dispositivos mediante el formato de carga útil de notación de objetos JavaScript (JSON), AWS IoT Core da formato a estos datos como JSON. Si utiliza el formato de carga de representación concisa de objetos binarios (CBOR), AWS IoT Core da formato a estos datos como CBOR. La `DeviceConfiguration` no admite objetos JSON anidados.

Funciones intrínsecas

Las funciones intrínsecas se utilizan en cualquier sección de la plantilla de aprovisionamiento, excepto en la sección `Mappings`.

`Fn::Join`

Añade un conjunto de valores a un único valor separado por el delimitador especificado. Si un delimitador es la cadena vacía, el conjunto de valores se concatena sin delimitador.

`Fn::Select`

Devuelve un único objeto de una lista de objetos por índice.

Important

`Fn::Select` no comprueba si hay valores `null` o si el índice queda fuera de los límites de la matriz. Ambas condiciones dan lugar a un error de aprovisionamiento, por lo que debe asegurarse de elegir un valor de índice válido y de que la lista contiene valores distintos de `null`.

`Fn::FindInMap`

Devuelve el valor correspondiente a claves en una asignación de dos niveles declarada en la sección `Mappings`.

`Fn::Split`

Divide una cadena en una lista de valores de cadena para que pueda seleccionar un elemento de la lista de cadenas. Especifique un delimitador que determine dónde se divide la cadena (por ejemplo, una coma). Después de dividir una cadena, utilice `Fn::Select` para seleccionar un elemento.

Por ejemplo, si una cadena delimitada por comas de IDs de subred se importa a la plantilla de pila, puede dividir la cadena en cada coma. En la lista de ID de subred, utilice `Fn::Select` para especificar el ID de subred de un recurso.

`Fn::Sub`

Sustituye variables en una cadena de entrada por los valores que especifique. Puede utilizar esta función para crear comandos o salidas que incluyan valores que no están disponibles hasta que crea o actualiza una pila.

Ejemplo de plantilla de aprovisionamiento de flotas

```
{  
    "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

Se puede actualizar una plantilla de aprovisionamiento existente para agregar un [enlace de preaprovisionamiento \(p. 816\)](#).

Enlaces de preaprovisionamiento

AWS recomienda utilizar funciones de enlaces de preaprovisionamiento al crear plantillas de aprovisionamiento para permitir un mayor control de qué dispositivos y cuántos dispositivos tiene incorporados su cuenta. Los enlaces de preaprovisionamiento son funciones de Lambda que validan los parámetros que se han pasado desde el dispositivo antes de permitir que se aprovisione el dispositivo. Esta función de Lambda debe existir en su cuenta antes de aprovisionar un dispositivo porque se llama cada vez que un dispositivo envía una solicitud a través de [the section called “RegisterThing” \(p. 823\)](#).

Important

Asegúrese de incluir `elsource-arn:source-account` en las claves de contexto de condición global de las políticas asociadas a la acción de Lambda para evitar la manipulación de permisos. Para obtener más información acerca de este tema, consulte [Prevención del suplente confuso entre servicios \(p. 343\)](#).

Para que los dispositivos se aprovisionen, la función Lambda debe aceptar el objeto de entrada y devolver el objeto de salida descrito en esta sección. El aprovisionamiento continúa solo si la función Lambda devuelve un objeto con `"allowProvisioning": True`.

Preaprovación de entrada de enlace

AWS IoT envía este objeto a la función Lambda cuando un dispositivo se registra con AWS 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",  
        ...  
    }  
}
```

El objeto `parameters` pasado a la función Lambda contiene las propiedades en el argumento `parameters` aprobado en la sección titulada ["RegisterThing" \(p. 823\)](#). Carga útil de solicitud.

Valor de retorno del enlace previo a la provisión

La función Lambda debe devolver una respuesta que indique si ha autorizado la solicitud de aprovisionamiento y los valores de las propiedades para anular.

A continuación se muestra un ejemplo de una respuesta exitosa de la función de preaprovación.

```
{  
    "allowProvisioning": true,  
    "parameterOverrides" : {  
        "Key": "newCustomValue",  
        ...  
    }  
}
```

"`parameterOverrides`" se añadirán valores a "`parameters`" parámetro de la sección titulada ["RegisterThing" \(p. 823\)](#). Carga útil de solicitud.

Note

- Si la función Lambda falla, la solicitud de aprovisionamiento falla con `ACCESS_DENIED` y se registra un error en CloudWatch Logs.
- Si la función Lambda no vuelve "`allowProvisioning": "true"` en la respuesta, la solicitud de aprovisionamiento falla con `ACCESS_DENIED`.
- La función Lambda debe terminar de ejecutarse y volver en 5 segundos; de lo contrario, la solicitud de aprovisionamiento falla.

Ejemplo de Lambda de enlace de preaprovación

Python

Un ejemplo de un enlace de preaprovación Lambda en Python.

```
import json
```

```
def pre_provisioning_hook(event, context):
    print(event)

    return {
        'allowProvisioning': True,
        'parameterOverrides': {
            'DeviceLocation': 'Seattle'
        }
    }
```

Java

Un ejemplo de un enlace de preaprovación Lambda en Java.

Clase de controlador:

```
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;
    }

}
```

Solicitud de clase:

```
package example;

import java.util.Map;
import lombok.Builder;
import lombok.Data;
import lombok.AllArgsConstructor;
import lombok.NoArgsConstructor;

@Data
@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;
}
```

Clase de respuesta:

```
package example;

import java.util.Map;
import lombok.Builder;
import lombok.Data;
import lombok.NoArgsConstructor;
import lombok.NonNullArgsConstructor;
import lombok.NoArgsConstructor;

@Data
@NoArgsConstructor
@NonFinalArgsConstructor
@NoArgsConstructorArgsConstructor
public class PreProvisioningHookResponse {
    private boolean allowProvisioning;
    private Map<String, String> parameterOverrides;
}
```

API de MQTT de aprovisionamiento de dispositivos

El servicio de aprovisionamiento de flotas admite estas API de MQTT:

- [the section called “CreateCertificateFromCsr” \(p. 819\)](#)
- [the section called “CreateKeysAndCertificate” \(p. 821\)](#)
- [the section called “RegisterThing” \(p. 823\)](#)

Esta API admite búferes de respuesta en formato de representación concisa de objetos binarios (CBOR) y notación de objetos JavaScript (JSON), dependiendo del [payload-format](#) del tema. Sin embargo, para una mayor claridad, los ejemplos de respuesta y solicitud de esta sección se muestran en formato JSON.

payload-format	Tipo de datos de formato de respuesta
cbor	Concise Binary Object Representation (Representación concisa de objetos binarios, CBOR)
json	JavaScript Object Notation (Notación de objetos de JavaScript, JSON)

Important

Antes de publicar un tema de mensaje de solicitud, suscríbase a los temas de respuesta para recibir la respuesta. Los mensajes utilizados por esta API utilizan el protocolo de publicación/suscripción de MQTT para proporcionar una interacción de solicitud y respuesta.

Si no se suscribe a los temas de respuesta antes de publicar una solicitud, es posible que no reciba los resultados de dicha solicitud.

CreateCertificateFromCsr

Crea un certificado a partir de una solicitud de firma de certificado (CSR). AWS IoT proporciona certificados de cliente firmados por la entidad de certificación (CA) Amazon Root. El nuevo certificado tiene un

estado PENDING_ACTIVATION. Cuando llama a RegisterThing para aprovisionar un objeto con este certificado, el estado del certificado cambia a la plantilla ACTIVE o INACTIVE tal como se describe en ella.

Note

Por seguridad, el certificateOwnershipToken devuelto por [CreateCertificateFromCsr \(p. 819\)](#) caduca después de una hora. [RegisterThing \(p. 823\)](#) debe llamarse antes de que el certificateOwnershipToken expire. Si el certificado creado por [CreateCertificateFromCsr \(p. 819\)](#) no se ha activado y no se ha adjuntado a una política o a una cosa en el momento en que caduca el token, se elimina el certificado. Si el token caduca, el dispositivo puede llamar [CreateCertificateFromCsr \(p. 819\)](#) para generar un nuevo certificado.

Solicitud CreateCertificateFromCsr

Publicar un mensaje con el tema \$aws/certificates/create-from-csr/[payload-format](#).

payload-format

El formato de carga del mensaje como cbor o json.

Carga de solicitud CreateCertificateFromCsr

```
{  
    "certificateSigningRequest": "string"  
}
```

certificateSigningRequest

La CSR, en formato PEM.

Respuesta CreateCertificateFromCsr

Suscripción a \$aws/certificates/create-from-csr/[payload-format](#)/accepted

payload-format

El formato de carga del mensaje como cbor o json.

Carga útil de respuesta CreateCertificateFromCsr

```
{  
    "certificateOwnershipToken": "string",  
    "certificateId": "string",  
    "certificatePem": "string"  
}
```

certificateOwnershipToken

El token para comprobar la propiedad del certificado durante el aprovisionamiento.

certificateId

El ID del certificado. Las operaciones de administración de certificados solo adoptan un certificateId.

certificatePem

Los datos del certificado, en formato PEM.

Error CreateCertificateFromCsr

Para recibir respuestas de error, suscríbase a `$aws/certificates/create-from-csr/payload-format/rejected`.

payload-format

El formato de carga del mensaje como cbor o json.

Carga de error CreateCertificateFromCsr

```
{  
    "statusCode": int,  
    "errorCode": "string",  
    "errorMessage": "string"  
}
```

statusCode

El código del estado.

errorCode

Código de error

errorMessage

Mensaje de error.

CreateKeysAndCertificate

Crea nuevas claves y un certificado. AWS IoT proporciona certificados de cliente firmados por la entidad de certificación (CA) Amazon Root. El nuevo certificado tiene un estado PENDING_ACTIVATION. Cuando llama a RegisterThing para aprovisionar un objeto con este certificado, el estado del certificado cambia a la plantilla ACTIVE o INACTIVE tal como se describe en ella.

Note

Por seguridad, el certificateOwnershipToken devuelto por [CreateKeysAndCertificate \(p. 821\)](#) caduca después de una hora. [RegisterThing \(p. 823\)](#) debe llamarse antes de que el certificateOwnershipToken expire. Si el certificado creado por [CreateKeysAndCertificate \(p. 821\)](#) no se ha activado y no se ha adjuntado a una política o a una cosa en el momento en que caduca el token, se elimina el certificado. Si el token caduca, el dispositivo puede llamar [CreateKeysAndCertificate \(p. 821\)](#) para generar un nuevo certificado.

Solicitud CreateKeysAndCertificate

Publicar un mensaje en `$aws/certificates/create/payload-format` con una carga de mensajes vacía.

payload-format

El formato de carga del mensaje como cbor o json.

Respuesta CreateKeysAndCertificate

Suscripción a \$aws/certificates/create/*payload-format*/accepted

payload-format

El formato de carga del mensaje como cbor o json.

Respuesta CreateKeysAndCertificate

```
{  
    "certificateId": "string",  
    "certificatePem": "string",  
    "privateKey": "string",  
    "certificateOwnershipToken": "string"  
}
```

certificateId

El ID del certificado.

certificatePem

Los datos del certificado, en formato PEM.

privateKey

La clave privada.

certificateOwnershipToken

El token para comprobar la propiedad del certificado durante el aprovisionamiento.

Error CreateKeysAndCertificate

Para recibir respuestas de error, suscríbase a \$aws/certificates/create/*payload-format*/rejected.

payload-format

El formato de carga del mensaje como cbor o json.

Carga de error CreateKeysAndCertificate

```
{  
    "statusCode": int,  
    "errorCode": "string",  
    "errorMessage": "string"  
}
```

statusCode

El código del estado.

errorCode
Código de error
errorMessage
Mensaje de error.

RegisterThing

Aprovisiona un objeto mediante una plantilla predefinida.

Solicitud RegisterThing

Publica un mensaje en \$aws/provisioning-templates/*templateName*/provision/*payload-format*.

payload-format
El formato de carga del mensaje como cbor o json.
templateName
El nombre de la plantilla de aprovisionamiento.

Carga útil de solicitud RegisterThing

```
{  
    "certificateOwnershipToken": "string",  
    "parameters": {  
        "string": "string",  
        ...  
    }  
}
```

certificateOwnershipToken

El token para comprobar la propiedad del certificado. El token lo genera AWS IoT cuando se crea un certificado a través de MQTT.

parameters

Opcional. Pares clave-valor del dispositivo que utilizan los [enlaces de preaprovisionamiento \(p. 816\)](#) para evaluar la solicitud de registro.

Respuesta de RegisterThing

Suscripción a \$aws/provisioning-templates/*templateName*/provision/*payload-format*/
accepted

payload-format
El formato de carga del mensaje como cbor o json.
templateName
El nombre de la plantilla de aprovisionamiento.

Carga útil de respuesta de RegisterThing

```
{  
    "deviceConfiguration": {  
        "string": "string",  
        ...  
    },  
    "thingName": "string"  
}
```

deviceConfiguration

La configuración del dispositivo definida en la plantilla.

thingName

El nombre del objeto de IoT creado durante el aprovisionamiento.

Respuesta de error de RegisterThing

Para recibir respuestas de error, suscríbase a `$aws/provisioning-templates/templateName/provision/payload-format/rejected`.

payload-format

El formato de carga del mensaje como `cbor` o `json`.

templateName

El nombre de la plantilla de aprovisionamiento.

Carga útil de respuesta de error de RegisterThing

```
{  
    "statusCode": int,  
    "errorCode": "string",  
    "errorMessage": "string"  
}
```

statusCode

El código del estado.

errorCode

Código de error

errorMessage

Mensaje de error.

Indexación de flotas

Note

La función de indexación de flotas para admitir la indexación denominada sombras y AWS IoT Device DefenderLos datos de infracciones están en versión preliminar para AWS IoT Device Managementy está sujeta a cambios.

Puede utilizar la indexación de flotas para indexar, buscar y agregar los datos de sus dispositivos de los siguientes orígenes:[AWS IoT Registro de \(p. 266\)](#),[AWS IoT Sombra de dispositivos \(p. 627\)](#),[AWS IoT Connectivity \(p. 1111\)](#), y[AWS IoT Device Defender \(p. 871\)](#)infracciones. Puede consultar un grupo de dispositivos y agregar estadísticas en registros de dispositivos que se basan en diferentes combinaciones de atributos de dispositivo, incluidos el estado, la conectividad y las infracciones del dispositivo. Con la indexación de flotas, puede organizar, investigar y solucionar problemas de su flota de dispositivos.

La indexación de flotas proporciona las siguientes funcionalidades.

- Administración de actualizaciones de índices

Puede configurar un índice de flota para que indexe las actualizaciones de sus grupos de objetos, registros de objetos, sombras de dispositivos, conectividad de dispositivos e infracciones de dispositivos. Cuando habilita la indexación de flotas,AWS IoT crea un índice para sus objetos o grupos de objetos.[AWS_Things](#)es el índice creado para todos los objetos.[AWS_ThingGroups](#)es el índice que contiene todos sus grupos de objetos. Después de activarla la indexación de flotas, puede realizar consultas en su índice, como buscar todos los dispositivos portátiles que tengan una duración de la batería de más del 70%.AWS IoT mantiene el índice continuamente actualizado con los datos más reciente. Para obtener más información, consulte[Administración de la indexación del \(p. 826\)](#).

- Búsqueda entre fuentes de datos

Puede crear una cadena de consulta basada en[un lenguaje de consulta sencillo \(p. 846\)](#)y úsalo para buscar en los orígenes de datos que configuras en la configuración de indexación de flotas. La cadena de consulta describe las cosas que desea encontrar. Para obtener más información acerca de las fuentes de datos que admiten la indexación de flotas, consulte[Administración de la indexación de objetos \(p. 829\)](#).

- Consulta de datos agregados

Puede buscar datos agregados y estadísticas de devolución en sus dispositivos, percentil, cardinalidad o una lista de cosas con consultas de búsqueda relacionadas con campos concretos. Para obtener más información acerca de la consulta de agregación, consulte[Consulta de datos agregados \(p. 840\)](#).

- Supervisión de datos agregados mediante métricas de flota

Puede utilizar métricas de flota para enviar automáticamente datos agregados a CloudWatch, analizar tendencias y crear alarmas para supervisar el estado agregado de su flota. Para obtener más información acerca de las métricas de flotas, consulte[Métricas de flota \(p. 851\)](#).

Para utilizar la indexación de flotas, debe configurar la configuración de indexación de flotas. Para configurar la configuración de indexación de flotas, puede utilizar el[AWS IoT consola](#), o si prefiere el acceso mediante programación, puede utilizar la AWSSDK o el AWS Command Line Interface(AWS CLI).

Para obtener información sobre los precios de este y otros servicios, consulte [AWS IoT Precios de Administración de dispositivos](#).

Administración de la indexación del

Note

La función de indexación de flotas para admitir la indexación denominada sombras y AWS IoT Device DefenderLos datos de infracciones están en versión preliminar para AWS IoT Device Management está sujeta a cambios.

La indexación de flotas administra dos tipos de índices por ti, la indexación de cosas y la indexación de grupos de cosas.

Indexación de elementos

El índice creado para todos los objetos es `AWS_Things`. La indexación de objetos es compatible con las siguientes fuentes de datos:[AWS IoT Registro de \(p. 266\)](#)datos,[AWS IoT Sombra de dispositivos \(p. 627\)](#)datos,[AWS IoT Connectivity \(p. 1111\)](#)datos, y[AWS IoT Device Defender \(p. 871\)](#)datos de infracciones.

Registro de-AWS IoTproporciona un registro que le ayuda a administrar los objetos. Con la indexación de flotas, puede añadir objetos al registro y buscar dispositivos. Para obtener más información acerca del registro, consulte.[Cómo administrar objetos con el registro \(p. 266\)](#).

Sombra-El[AWS IoT Servicio Device Shadow de \(p. 627\)](#)proporciona sombras que le ayudan a almacenar los datos de estado de sus dispositivos. La indexación de cosas admite sombras clásicas sin nombre y sombras con nombre. Para obtener más información acerca de las sombras, consulte.[AWS IoT Servicio Device Shadow de \(p. 627\)](#).

Conectividad-Los datos de conectividad del dispositivo le ayudan a identificar el estado de conexión de sus dispositivos. Estos datos de conectividad están impulsados por[Eventos del ciclo de vida \(p. 1111\)](#). Cuando un cliente se conecta o se desconecta,AWS IoTpublica los eventos del ciclo de vida con mensajes en los temas de MQTT. Un mensaje de conexión o desconexión puede ser una lista de elementos JSON que proporcionan detalles del estado de la conexión. Para obtener más información acerca de la conexión de dispositivos, consulte.[Eventos del ciclo de vida \(p. 1111\)](#).

Infracciones del Defender-AWS IoT Device Defenderlos datos de infracciones ayudan a identificar los comportamientos anómalos de sus dispositivos con respecto a los comportamientos normales que define en un perfil de seguridad. Un perfil de seguridad contiene un conjunto de comportamientos de dispositivos esperados. Cada comportamiento utiliza una métrica que especifica el comportamiento normal de los dispositivos. Para obtener más información sobre las infracciones de Device Defender, consulte[AWS IoT Device Defender Detectar \(p. 983\)](#).

Para obtener más información, consulte[Administración de la indexación de objetos \(p. 829\)](#).

Indexación de grupos de elementos

`AWS_ThingGroups` es el índice que contiene todos sus grupos de objetos. Puede usar este índice para buscar grupos en función de su nombre, descripción, atributos y todos los nombres de grupos principales.

Para obtener más información, consulte[Administración de la indexación de grupos de objetos \(p. 839\)](#).

Campos administrados

Los campos administrados contienen datos asociados con objetos, grupos de objetos, sombras de dispositivos, conectividad de dispositivos e infracciones de Device Defender. AWS IoT define el tipo de datos en los campos administrados. El usuario especifica los valores de cada campo administrado cuando se crea un objeto de IoT. Por ejemplo, los nombres de objetos, los grupos de objetos y las descripciones de objetos son todos campos administrados. La indexación de flotas indexa los campos administrados en función del modo de indexación que especifique. Los campos administrados no se pueden cambiar ni pueden aparecer en `customFields`. Para obtener más información, consulte [Campos personalizados \(p. 828\)](#).

A continuación se enumeran los campos administrados para la indexación de cosas:

- Campos administrados del 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 administrados de sombras clásicas sin nombre

```
"managedFields" : [
    {name:shadow.version, type:Number},
    {name:shadow.hasDelta, type:Boolean}
]
```

- Campos administrados de sombras con nombre

```
"managedFields" : [
    {name:shadow.name.shadowName.version, type:Number},
    {name:shadow.name.shadowName.hasDelta, type:Boolean}
]
```

- Campos administrados de conectividad de objetos

```
"managedFields" : [
    {name:connectivity.timestamp, type:Number},
    {name:connectivity.version, type:Number},
    {name:connectivity.connected, type:Boolean},
    {name:connectivity.disconnectReason, type:String}
]
```

- Campos administrados 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 administrados de grupos de objetos

```
"managedFields" : [
```

```
{name:description, type:String},  
{name:parentGroupNames, type:String},  
{name:thingGroupId, type:String},  
{name:thingGroupName, type:String},  
{name:version, type:Number},  
]
```

En la tabla siguiente se enumeran los campos administrados que no se pueden realizar búsquedas.

Origen de datos	Campo administrado que no se puede realizar búsquedas
Registry (Registro)	registry.version
Sombras sin nombre	shadow.version
Sombras con nombre	shadow.name.*.version
Device Defender	deviceDefender.version
Grupos de elementos	version

Campos personalizados

Puede agregar atributos de cosa, datos de Device Shadow y datos de infracciones de Device Defender creando campos personalizados para indexarlos. La `customFields` atributo es una lista de pares de nombres de campo y tipos de datos. Puede realizar consultas de agregación según el tipo de datos. El modo de indexación que elija afecta a los campos que se pueden especificar en `customFields`. Por ejemplo, si especifica la `REGISTRY` modo de indexación, no puede especificar un campo personalizado de una sombra de objeto. Puede utilizar el [configuración de indexación de actualización](#) Comando CLI para crear o actualizar los campos personalizados (consulte un comando de ejemplo en [Actualización de ejemplos de configuración de indexación \(p. 831\)](#)).

- Nombres de campos personalizados

Los nombres de campo personalizados para los atributos de grupos de cosas y cosas comienzan por `attributes.`, seguido del nombre del atributo. Si la indexación de sombras sin nombre está activada, las cosas pueden tener nombres de campo personalizados que empiecen por `shadow.desired.shadow.reported`, seguido del nombre del valor de datos de sombra sin nombre. Si la indexación de sombras con nombre está activada, las cosas pueden tener nombres de campo personalizados que empiecen por `shadow.name.*.desired.shadow.name.*.reported`, seguido del valor de datos de sombra con nombre. Si la indexación de infracciones de Device Defender está activada, las cosas pueden tener nombres de campo personalizados que empiecen por `deviceDefender`, seguido del valor de datos de infracciones de Device Defender.

El nombre del atributo o valor de datos que sigue al prefijo solo puede contener caracteres alfanuméricos, - (guión) y _ (guión bajo). No puede tener espacios.

Si el tipo del campo personalizado de la configuración y el valor que se va indexar no coinciden, la indexación de flotas omite el valor no coincidente para las consultas de agregación. Los registros de CloudWatch son útiles a la hora de solucionar problemas de consultas de agregación. Para obtener más información, consulte [Solución de problemas de consultas de agregación en el servicio de indexación de flotas \(p. 1284\)](#).

- Tipos de campos personalizados

Los tipos de campo personalizados tienen los siguientes valores admitidos: `Number`, `String`, y `Boolean`.

Administración de la indexación de objetos

Note

La función de indexación de flotas para admitir la indexación denominada sombras y AWS IoT Device Defender Los datos de infracciones están en versión preliminar para AWS IoT Device Management y está sujeta a cambios.

El índice creado para todos los objetos es `AWS_Things`. Puede controlar qué indexar desde los siguientes orígenes de datos: [AWS IoT Registro de \(p. 266\)](#) datos, [AWS IoT Sombra de dispositivos \(p. 627\)](#) datos, [AWS IoT Connectivity \(p. 1111\)](#) datos, y [AWS IoT Device Defender \(p. 871\)](#) datos de infracciones.

Habilitación de la indexación de objetos

Usa el [configuración de indexación de actualización](#) CLI command o [el UpdateIndexingConfiguration](#) Operación de API para crear el `AWS_Things` indexar y controlar su configuración. Mediante el uso de la `--thing-indexing-configuration(thingIndexingConfiguration)`, controla qué tipo de datos (por ejemplo, registro, sombra, datos de conectividad de dispositivos y datos de infracciones de Device Defender) están indexados.

El parámetro `--thing-indexing-configuration` toma una cadena con la siguiente estructura:

```
{  
    "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 indexación de elementos

El atributo `thingIndexingMode` controla qué tipo de datos se indexan.

Important

Para habilitar la indexación de cosas, el `thingIndexingMode` se puede configurar para que esté DESACTIVADO.

Atributo	Valores válidos	Descripción
<code>thingIndexingMode</code>	OFF	Sin indexación.

Atributo	Valores válidos	Descripción
	REGISTRY	Indexar los datos de registro.
	REGISTRY_AND_SHADOW	Indexar los datos de registro y de sombras de objetos.

El atributo `thingConnectivityIndexingMode` especifica si los datos de conectividad de objetos están indexados.

Atributo	Valores válidos	Descripción
<code>thingConnectivityIndexingMode</code>	No especificado.	Lo que los datos de conectividad no están indexados.
	OFF	Lo que los datos de conectividad no están indexados.
	STATUS	Los datos de conectividad están indexados.

La `deviceDefenderIndexingMode` especifica si los datos de infracciones de Device Defender están indexados.

Atributo	Valores válidos	Descripción
<code>deviceDefenderIndexingMode</code>	No especificado.	Los datos de infracciones de Device Defender no están indexados.
	OFF	Los datos de infracciones de Device Defender no están indexados.
	INFRACCIONES	Los datos de infracciones de Device Defender están indexados.

La `namedShadowIndexingMode` atributo especifica si los datos de sombra con nombre están indexados.

Atributo	Valores válidos	Descripción
<code>namedShadowIndexingMode</code>	No especificado.	Los datos de sombra con nombre no están indexados.
	OFF	Los datos de sombra con nombre no están indexados.
	ACTIVAR	Los datos de sombra con nombre se indexan.

Campos administrados y campos personalizados

Campos administrados

Los campos administrados contienen datos asociados con objetos, grupos de objetos, sombras de dispositivos, conectividad de dispositivos e infracciones de Device Defender. AWS IoT define el tipo de datos en los campos administrados. El usuario especifica los valores de cada campo administrado cuando se crea un objeto de IoT. Por ejemplo, los nombres de objetos, los grupos de objetos y las descripciones de objetos son todos campos administrados. La indexación de flotas indexa los campos administrados en función del modo de indexación que especifique. Los campos administrados no se pueden cambiar ni pueden aparecer en `customFields`.

Campos personalizados

Puede agregar atributos, datos de Device Shadow y datos de infracciones de Device Defender creando campos personalizados para indexarlos. El atributo `customFields` es una lista de pares de nombres de campo y tipos de datos. Puede realizar consultas de agregación según el tipo de datos. El modo de indexación que elija afecta a los campos que se pueden especificar en `customFields`. Por ejemplo, si especifica el modo `REGISTRY` de indexación, no puede especificar un campo personalizado de una sombra de objeto. Puede utilizar el comando `update-indexing-configuration` para crear o actualizar los campos personalizados (consulte un comando de ejemplo en [Actualización de ejemplos de configuración de indexación \(p. 831\)](#)). Para obtener más información, consulte [Campos personalizados](#) (p. 828).

Actualización de ejemplos de configuración de indexación

Puede utilizar el comando `update-indexing-configuration` de la CLI de AWS IoT para actualizar la configuración de indexación. En los siguientes ejemplos se muestra cómo usar `update-indexing-configuration`.

Sintaxis corta:

```
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}]'
```

Sintaxis de 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"} ] }
```

Este comando no produce ningún resultado.

Para comprobar el estado del índice de cosas, ejecute la comando `describe-index` de la CLI:

```
aws iot describe-index --index-name "AWS_Things"
```

El resultado del comando `describe-index` tendrá un aspecto similar al siguiente:

```
{  
    "indexName": "AWS_Things",  
    "indexStatus": "ACTIVE",  
    "schema": "MULTI_INDEXING_MODE"  
}
```

Note

La indexación de flotas puede tardar un momento en actualizar el índice de flota. Recomendamos esperar hasta que el indexStatus muestra ACTIVE antes de usarlo. Puede tener distintos valores en el campo de esquema según las fuentes de datos que haya configurado. Para obtener más información, consulte [Descripción de un índice de objeto \(p. 834\)](#).

Para obtener los detalles de la configuración de indexación de tus cosas, ejecuta el `get-indexing-configuration` de la CLI:

```
aws iot get-indexing-configuration
```

El resultado del comando `get-indexing-configuration` tendrá un aspecto similar al siguiente:

```
{  
    "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 actualizar los campos personalizados, puede ejecutar el comando `update-indexing-configuration`. A continuación se indica un ejemplo:

```

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}]'

```

Este comando se añadió `shadow.desired.intensity` a la configuración de indexación.

Note

La actualización de la configuración de indexación de campos personalizados sobrescribe todos los campos personalizados existentes. Asegúrese de especificar todos los campos personalizados cuando llame a `update-indexing-configuration`.

Después de reconstruir el índice, puede utilizar una consulta de agregación en los campos recién añadidos, los datos de registro de búsqueda, los datos de sombra y los datos de estado de conectividad de objetos.

Al cambiar el modo de indexación, asegúrese de que todos los campos personalizados son válidos mediante el nuevo modo de indexación. Por ejemplo, si empiezas a usar `REGISTRY_AND_SHADOW` modo con un campo personalizado llamado `shadow.desired.temperature`, debe eliminar `elshadow.desired.temperature` campo personalizado antes de cambiar el modo de indexación a `REGISTRY`. Si la configuración de indexación contiene campos personalizados que no están indexados por el modo de indexación, se producirá un error en la actualización.

Descripción de un índice de objeto

El comando siguiente muestra cómo utilizar el comando `describe-index` de la CLI para recuperar el estado actual del índice del objeto:

```
aws iot describe-index --index-name "AWS_Things"
```

La respuesta del comando puede ser similar a la siguiente:

```
{  
    "indexName": "AWS_Things",  
    "indexStatus": "BUILDING",  
    "schema": "REGISTRY_AND_SHADOW_AND_CONNECTIVITY_STATUS"  
}
```

La primera vez que habilita la indexación de flotas, AWS IoT crea tu índice. Cuando `indexStatus` está en el `BUILDING` estado, no se puede consultar el índice. El valor `schema` del índice de objetos indica qué tipo de datos (`REGISTRY_AND_SHADOW_AND_CONNECTIVITY_STATUS`) se indexan.

Si se cambia la configuración del índice, el índice se vuelve a compilar. Durante este proceso, `indexStatus` es `REBUILDING`. Puede ejecutar consultas en los datos del índice de objetos mientras se está generando. Por ejemplo, si cambia la configuración del índice de `REGISTRY` a `REGISTRY_AND_SHADOW`, cuando el índice se vuelve a generar, puede consultar los datos del registro, incluidas las últimas actualizaciones. Sin embargo, no puede consultar los datos de sombra hasta que se complete la recompilación. El tiempo que tarda en compilar o recompilar el índice depende de la cantidad de datos.

Puede ver distintos valores en el campo de esquema en función de los orígenes de datos que haya configurado. En la tabla siguiente se muestran los distintos valores de esquema y las descripciones correspondientes:

Esquema	Descripción
OFF	No se configuran ni indexan fuentes de datos.
REGISTRY	Los datos del registro están indexados.
REGISTRY_AND_SHADOW	Se indexan los datos del registro y los datos de sombra sin nombre (clásicos).
REGISTRY_AND_CONNECTIVITY	Los datos de registro y los datos de conectividad se indexan.
REGISTRY_AND_SHADOW_AND_CONNECTIVITY	Se indexan los datos del registro, los datos de sombra sin nombre (clásicos) y los datos de conectividad se indexan.

Esquema	Descripción
MODO MULTI_INDEXING_	Los datos de infracciones de sombra o Device Defender con nombre se indexan, además de datos de conectividad o sombra sin nombre (clásicos) de registro.

Consulta de un índice de objeto

Puede utilizar el comando search-index de la CLI para consultar los datos del 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"
}
```

En la respuesta JSON, "connectivity" (según lo habilitado por `elthingConnectivityIndexingMode=STATUS`) proporciona un valor booleano, una marca temporal y una `DisconnectReason` que indica si el dispositivo está conectado a AWS IoT Core. El dispositivo "mything1" desconexión (`false`) en el momento POSIX 1556649874716 debido a `CONNECTION_LOST`. Para obtener más información acerca de los motivos de desconexión, consulte [Eventos del ciclo de vida \(p. 1111\)](#).

```
"connectivity": {
    "connected": false,
    "timestamp": 1556649874716,
    "disconnectReason": "CONNECTION_LOST"
}
```

El dispositivo "mything2" conectado (`true`) en tiempo POSIX 1556649855046:

```
"connectivity": {
    "connected": true,
    "timestamp": 1556649855046
}
```

Las marcas temporales se indican en milisegundos desde la fecha de inicio, por lo que 1556649855046 representa 6:44:15 .046 p. m. el martes 30 de abril de 2019 (UTC).

Important

Si un dispositivo se ha desconectado durante aproximadamente una hora, el "timestamp" Value y el "disconnectReason" Puede que no aparezca el valor del estado de conectividad.

Restricciones y limitaciones

Estas son las restricciones y limitaciones de AWS_Things.

Campos de sombra con tipos complejos

Un campo de sombra se indexa solo si el valor del campo es un tipo sencillo, como un objeto JSON que no incluya una matriz o una matriz que consta en su totalidad de tipos sencillos. Un tipo sencillo es una cadena, un número o uno de los literales `true` o `false`. Por ejemplo, dado el siguiente estado de sombra, el valor del campo "palette" no se indexa porque es una matriz que contiene elementos de tipos complejos. El valor del campo "colors" sí se indexará porque cada valor de la matriz es una cadena.

```
{  
    "state": {  
        "reported": {  
            "switched": "ON",  
            "colors": [ "RED", "GREEN", "BLUE" ],  
            "palette": [  
                {  
                    "name": "RED",  
                    "intensity": 124  
                },  
                {  
                    "name": "GREEN",  
                    "intensity": 68  
                },  
                {  
                    "name": "BLUE",  
                    "intensity": 201  
                }  
            ]  
        }  
    }  
}
```

Nombres de campos de sombra anidados

Los nombres de los campos de sombra anidados se almacenan como una cadena delimitada por punto (.). Por ejemplo, dado un documento de sombra:

```
{  
    "state": {  
        "desired": {  
            "one": {  
                "two": {  
                    "three": "v2"  
                }  
            }  
        }  
    }  
}
```

Nombre del campo `dethreee` almacena como `desired.one.two.three`. Si también tiene un documento de sombra, se almacena así:

```
{  
    "state": {  
        "desired": {  
            "one.two.three": "v2"  
        }  
    }  
}
```

coinciden con una consulta `deshadow.desired.one.two.three:v2`. La práctica recomendada es no utilizar puntos en los nombres de campos de sombra.

Metadatos de sombra

Un campo en una sección de metadatos de sombra se indexa, pero solo si se indexa el campo correspondiente en la sección "state" de la sombra. (En el ejemplo anterior, la "palette" en la sección de metadatos de la sombra tampoco se indexa).

Sombras no registradas

Si usa [UpdateThingShadow](#) para crear una sombra mediante un nombre de objeto que no se ha registrado en su AWS IoT cuenta, los campos de esta sombra no están indexados. Esto se aplica tanto a las sombras clásicas sin nombre como a las sombras con nombre.

Valores numéricos

Si el servicio reconoce como valor numérico algún dato de sombra o de registro, se indexa como tal. Puede formar consultas que impliquen rangos y operadores de comparación en valores numéricos (por ejemplo "attribute.foo<5" o "shadow.reported.foo:[75 TO 80]"). Para que se reconozca como numérico, el valor de los datos debe ser un número JSON de tipo literal válido. El valor puede ser un entero en el rango -2⁵³... 2⁵³-1, un punto flotante de doble precisión con notación exponencial opcional o parte de una matriz que solo contenga estos valores.

Valores nulos

Los valores nulos no están indexados.

Valores máximos

El número máximo de campos personalizados para consultas de agregación es 5.

El número máximo de percentiles solicitados para consultas de agregación es 100.

El tamaño total de datos máximo de una cosa procesada por la indexación de flotas está limitado a 32 KB. Estos datos incluyen datos indexados del registro, sombras clásicas y con nombre, eventos del ciclo de vida de conectividad y datos de infracciones de Device Defender.

El número máximo de sombras con nombre por objeto es 5.

El ancho de banda máximo que admite la indexación de flotas es de 32 MBps.

Autorización

Puede especificar el índice de objetos como un nombre de recurso de Amazon (ARN) en un AWS IoT medidas políticas, según se indica a continuación.

Acción	Recurso
<code>iot:SearchIndex</code>	El ARN de un índice (por ejemplo, <code>arn:aws:iot:your-aws-region:your-aws-account:index/AWS_Things</code>).
<code>iot:DescribeIndex</code>	El ARN de un índice (por ejemplo, <code>arn:aws:iot:your-aws-region:index/AWS_Things</code>).

Note

Si tiene los permisos para consultar el índice de la flota, podrá obtener acceso a los datos de los objetos en toda la flota.

Administración de la indexación de grupos de objetos

`AWS_ThingGroups` es el índice que contiene todos sus grupos de objetos. Puede usar este índice para buscar grupos en función de su nombre, descripción, atributos y todos los nombres de grupos principales.

Habilitación de la indexación de grupos de objetos

Puede utilizar el valor `thing-group-indexing-configuration` en la API [UpdateIndexingConfiguration](#) para crear el índice `AWS_ThingGroups` y controlar su configuración. Puede utilizar la API [GetIndexingConfiguration](#) para recuperar la configuración de indexación actual.

Para actualizar las configuraciones de indexación de grupos de objetos, ejecute la `update-indexing-configurationCommand` de la CLI:

```
aws iot update-indexing-configuration --thing-group-indexing-configuration  
thingGroupIndexingMode=ON
```

También puede actualizar las configuraciones para la indexación de objetos y grupos de objetos en un único comando, como en el siguiente ejemplo:

```
aws iot update-indexing-configuration --thing-indexing-configuration  
thingIndexingMode=REGISTRY --thing-group-indexing-configuration thingGroupIndexingMode=ON
```

Los siguientes valores son válidos para `thingGroupIndexingMode`.

OFF

Sin indexación/eliminación del índice.

ACTIVAR

Cree o configure el índice `AWS_ThingGroups`.

Para recuperar las configuraciones de indexación actuales de objetos y grupos de objetos, ejecute la `get-indexing-configurationCommand` de la CLI:

```
aws iot get-indexing-configuration
```

La respuesta del comando tendrá un aspecto similar al siguiente:

```
{  
    "thingGroupIndexingConfiguration": {  
        "thingGroupIndexingMode": "ON"  
    }  
}
```

Descripción de índices de grupos

Para recuperar el estado actual del `AWS_ThingGroups` index, utilice el `describe-indexCommand` de la CLI:

```
aws iot describe-index --index-name "AWS_ThingGroups"
```

La respuesta del comando tendrá un aspecto similar al siguiente:

```
{
```

```
    "indexStatus": "ACTIVE",
    "indexName": "AWS_ThingGroups",
    "schema": "THING_GROUPS"
}
```

AWS IoT crea el índice cuando se habilita la indexación por primera vez. No se puede consultar el índice si `indexStatus` es `BUILDING`.

Consulta de un índice de grupo de objetos

Para consultar los datos del índice, utilice `lsearch-indexCommand` de la CLI:

```
aws iot search-index --index-name "AWS_ThingGroups" --query-string
"thingGroupName:mythinggroup*"
```

Autorización

Puede especificar el índice de grupos de objetos como un ARN del recurso en una acción de política de AWS IoT, como en el siguiente ejemplo.

Acción	Recurso
<code>iot:SearchIndex</code>	El ARN de un índice (por ejemplo, <code>arn:aws:iot:your-aws-region:index/AWS_ThingGroups</code>).
<code>iot:DescribeIndex</code>	El ARN de un índice (por ejemplo, <code>arn:aws:iot:your-aws-region:index/AWS_ThingGroups</code>).

Consulta de datos agregados

AWS IoT proporciona cuatro API (`GetStatistics`, `GetCardinality`, `GetPercentiles`, `yGetBucketsAggregation`) que le permiten buscar datos agregados en su flota de dispositivos.

Note

En caso de problemas con valores faltantes o inesperados para las API de agregación, lea [Guía de solución de problemas de indexación de flotas \(p. 1284\)](#).

GetStatistics

La API `GetStatistics` y el comando `get-statistics` de la CLI devuelven el recuento, la media, la suma, el mínimo, el máximo, la suma de los cuadrados, la varianza y la desviación estándar del campo agregado especificado.

El comando `get-statistics` de la CLI usa los siguientes parámetros:

`index-name`

El nombre del índice que se buscará. El valor predeterminado es `AWS_Things`.

`query-string`

La consulta utilizada para buscar el índice. Puede especificar "*" para obtener el recuento de todos los objetos indexados en su cuenta de AWS.

aggregationField

Opcional. El campo que se va a agregar. Este campo debe ser un campo administrado o personalizado definido al llamar a update-indexing-configuration. Si no especifica un campo de agregación, se utiliza `registry.version` como campo de agregación.

query-version

La versión de la consulta que se va a utilizar. El valor predeterminado es 2017-09-30.

El tipo de campo de agregación puede afectar a las estadísticas devueltas.

GetStatistics con valores de cadena

Si realiza la agregación en un campo de cadena, la llamada a `GetStatistics` devuelve el número de dispositivos que tienen atributos que coinciden con la consulta. Por ejemplo:

```
aws iot get-statistics --aggregation-field 'attributes.stringAttribute'  
--query-string '*'
```

Este comando devuelve el número de dispositivos que contienen un atributo llamado `stringAttribute`:

```
{  
  "statistics": {  
    "count": 3  
  }  
}
```

GetStatistics con valores booleanos

Cuando llama a `GetStatistics` con un campo de agregación booleano:

- `AVERAGE` es el porcentaje de dispositivos que coinciden con la consulta.
- `MINIMUM` es 0 o 1 de acuerdo con las siguientes reglas:
 - Si todos los valores del campo de agregación son `false`, `MINIMUM` es 0.
 - Si todos los valores del campo de agregación son `true`, `MINIMUM` es 1.
 - Si los valores del campo de agregación son una mezcla de `false` y `true`, `MINIMUM` es 0.
- `MAXIMUM` es 0 o 1 de acuerdo con las siguientes reglas:
 - Si todos los valores del campo de agregación son `false`, `MAXIMUM` es 0.
 - Si todos los valores del campo de agregación son `true`, `MAXIMUM` es 1.
 - Si los valores del campo de agregación son una mezcla de `false` y `true`, `MAXIMUM` es 1.
- `SUM` es la suma del entero equivalente de los valores booleanos.
- `COUNT` es el recuento de elementos que coinciden con los criterios de cadena de consulta y contienen un valor de campo de agregación válido.

GetStatistics con valores numéricos

Cuando se llama a `GetStatistics` y se especifica un campo de agregación de tipo `Number`, `GetStatistics` devuelve los siguientes valores:

count

El número de objetos que coinciden con los criterios de cadena de consulta y contienen un valor de campo de agregación válido.

average

El promedio de los valores numéricos que coinciden con la consulta.

sum

La suma de los valores numéricos que coinciden con la consulta.

minimum

El menor de los valores numéricos que coinciden con la consulta.

maximum

El mayor de los valores numéricos que coinciden con la consulta.

sumOfSquares

La suma de los cuadrados de los valores numéricos que coinciden con la consulta.

variance

La varianza de los valores numéricos que coinciden con la consulta. La varianza de un conjunto de valores es la media de los cuadrados de las diferencias de cada valor con respecto al valor medio del conjunto.

stdDeviation

La desviación estándar de los valores numéricos que coinciden con la consulta. La desviación estándar de un conjunto de valores es una medida de la distribución de los valores.

El siguiente ejemplo muestra cómo llamar a get-statistics con un campo numérico personalizado.

```
aws iot get-statistics --aggregation-field 'attributes.numericAttribute2'  
--query-string '*'
```

```
{  
  "statistics": {  
    "count": 3,  
    "average": 33.33333333333336,  
    "sum": 100.0,  
    "minimum": -125.0,  
    "maximum": 150.0,  
    "sumOfSquares": 43750.0,  
    "variance": 13472.22222222222,  
    "stdDeviation": 116.06990230986766  
  }  
}
```

Para los campos de agregación numérica, si los valores del campo superan el valor "double" máximo, los valores de las estadísticas están vacíos

GetCardinality

La API [GetCardinality](#) y el comando get-cardinality de la CLI devuelven el recuento aproximado de valores únicos que coinciden con la consulta. Por ejemplo, es posible que desee encontrar el número de dispositivos con niveles de batería inferiores al 50 por ciento:

```
aws iot get-cardinality --index-name AWS_Things --query-string "batteryLevel  
> 50" --aggregation-field "shadow.reported.batteryLevel"
```

Este comando devuelve el número de objetos con niveles de batería de más del 50 por ciento:

```
{  
    "cardinality": 100  
}
```

get-cardinality siempre devuelve cardinality, aunque no haya campos coincidentes. Por ejemplo:

```
aws iot get-cardinality --query-string "thingName:Non-existent*"  
--aggregation-field "attributes.customField_STR"
```

```
{  
    "cardinality": 0  
}
```

El comando get-cardinality de la CLI usa los siguientes parámetros:

index-name

El nombre del índice que se buscará. El valor predeterminado es AWS_Things.

query-string

La consulta utilizada para buscar el índice. Puede especificar "*" para obtener el recuento de todos los objetos indexados en su cuenta de AWS.

aggregationField

El campo que se va a agregar.

query-version

La versión de la consulta que se va a utilizar. El valor predeterminado es 2017-09-30.

GetPercentiles

La API [GetPercentiles](#) y el comando get-percentiles de la CLI agrupa los valores agregados que coinciden con la consulta en grupos de percentiles. Los grupos de percentiles predeterminados son: 1,5,25,50,75,95,99, aunque puede especificar los suyos propios cuando llame a GetPercentiles. Esta función devuelve un valor para cada grupo de percentiles especificado (o para los grupos de percentiles predeterminados). El grupo de percentiles "1" contiene el valor agregado del campo que se obtiene aproximadamente en el uno por ciento de los valores que coinciden con la consulta. El grupo de percentiles "5" contiene el valor agregado del campo que se obtiene en aproximadamente el cinco por ciento de los valores que coinciden con la consulta, y así sucesivamente. El resultado es una aproximación: cuantos más valores coincidan con la consulta, más precisos serán los valores de percentil.

El siguiente ejemplo muestra cómo llamar al comando get-percentiles de la 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": 99.0  
        }  
    ]  
}
```

```
        "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
    }
]
}
```

El siguiente comando muestra la salida devuelta get-percentiles cuando no hay documentos coincidentes.

```
aws iot get-percentiles --query-string "thingName:Non-existent*"
--aggregation-field "attributes.customField_NUM"
```

```
{
    "percentiles": []
}
```

El comando get-percentile de la CLI usa los siguientes parámetros:

index-name

El nombre del índice que se buscará. El valor predeterminado es AWS_Things.

query-string

La consulta utilizada para buscar el índice. Puede especificar "*" para obtener el recuento de todos los objetos indexados en su cuenta de AWS.

aggregationField

El campo que se va a agregar, que debe ser del tipo Number.

query-version

La versión de la consulta que se va a utilizar. El valor predeterminado es 2017-09-30.

percents

Opcional. Puede utilizar este parámetro para especificar grupos de percentiles personalizados.

Agregación GetBuckets

La [Agregación GetBuckets](#) API y el `get-buckets-aggregation` comando CLI devuelven una lista de depósitos y el número total de elementos que se ajustan a los criterios de cadena de consulta.

El siguiente ejemplo muestra cómo llamar al comando de la 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}'
```

Este comando devuelve lo siguiente:

```
{  
    "totalCount": 20,  
    "buckets": [  
        {  
            "keyValue": "100",  
            "count": 12  
        },  
        {  
            "keyValue": "90",  
            "count": 5  
        },  
        {  
            "keyValue": "75",  
            "count": 3  
        }  
    ]  
}
```

El comando de la CLI `get-buckets-aggregation` usa los siguientes parámetros:

index-name

El nombre del índice que se buscará. El valor predeterminado es `AWS_Things`.

query-string

La consulta utilizada para buscar el índice. Puede especificar "*" para obtener el recuento de todos los objetos indexados en su cuenta de AWS.

aggregation-field

El campo que se va a agregar.

buckets-aggregation-type

Control básico de la forma de respuesta y del tipo de agregación de bucket que se va a realizar.

Autorización

Puede especificar el índice de grupos de objetos como un ARN del recurso en una acción de política de AWS IoT, como en el siguiente ejemplo.

Acción	Recurso
iot:GetStatistics	El ARN de un índice (por ejemplo, arn:aws:iot: your-aws-region :index/AWS_Things o arn:aws:iot: your-aws-region :index/AWS_ThingGroups).

Sintaxis de la consulta

En la indexación de flotas, se utiliza una sintaxis de consulta para especificar consultas.

Características admitidas

La sintaxis de consulta es compatible con las siguientes características:

- Términos y frases
- Búsqueda de campos
- Búsqueda de prefijos
- Búsqueda de intervalos
- Operadores booleanos AND, OR, NOT y -. El guion se utiliza para excluir algo de los resultados de la búsqueda (por ejemplo, thingName:(tv* AND -plasma)).
- Agrupación
- Agrupación de campos
- Escapar caracteres especiales (por ejemplo, con \)

Características de no admitidas

La sintaxis de consulta no admite las siguientes características de:

- Uso de caracteres comodín al principio de la búsqueda (como "*xyz"), pero si se utiliza "*" en la búsqueda se devolverán todos los objetos
- Expresiones regulares
- Impulso
- Clasificación
- Búsquedas confusas
- Búsqueda de proximidad
- Ordenar
- Agregación

Notas

Algunas cosas que tener en cuenta acerca del idioma de consulta:

- El operador predeterminado es AND. La consulta "thingName:abc thingType:xyz" es igual que "thingName:abc AND thingType:xyz".
- Si no se especifica un campo, AWS IoT busca el término en todos los campos Registro, Device Shadow y Device Defender.

- Todos los nombres de campo distinguen entre mayúsculas y minúsculas.
- La búsqueda distingue entre mayúsculas y minúsculas. Las palabras están separadas por caracteres de espacio en blanco, tal como define de `JavaCharacter.isWhitespace(int)`.
- La indexación de datos de sombra de dispositivo (sombras sin nombre y sombras con nombre) incluye secciones informadas, deseadas, delta y de metadatos.
- Las versiones de sombra y registro del dispositivo no permiten búsquedas, pero están presentes en la respuesta.
- El número máximo de términos en una consulta es siete.

Ejemplo de consultas de objetos

Note

La función de indexación de flotas para admitir la indexación denominada sombras y AWS IoT Device DefenderLos datos de infracciones están en versión preliminar para AWS IoT Device Managementy está sujeta a cambios.

Especifique consultas en una cadena de consulta mediante una sintaxis de consulta. Las consultas se pasan al [SearchIndex API](#). En la siguiente tabla se enumeran algunas cadenas de consulta de ejemplo.

Cadena de consulta	Resultado
abc	Consultas de «abc» en cualquier campo de registro, sombra (sombra clásica sin nombre y sombra con nombre) o infracciones de Device Defender.
thingName:myThingName	Consulta un objeto con el nombre "myThingName".
thingName:my*	Consulta los objetos cuyos nombres que comienzan por "my".
thingName:ab?	Consulta los objetos cuyos nombres tienen «ab» además de un carácter adicional (por ejemplo, «aba», «abb», «abc», etc.)
thingTypeName:aa	Las consultas de objetos que están asociados con el tipo «aa».
attributes.myAttribute:75	Consulta los objetos con un atributo denominado "myAttribute" que tiene el valor 75.
attributes.myAttribute:[75 TO 80]	Consulta los objetos con un atributo denominado «MyAttribute» que tiene un valor que se encuentra dentro de un rango numérico (entre 75 y 80, ambos incluidos).
attributes.myAttribute:{75 TO 80}	Consulta los objetos con un atributo denominado «MyAttribute» que tiene un valor que se encuentra dentro del rango numérico (>75 y <=80).
attributes.serialNumber:["abcd" TO "abcdef"]	Consulta los objetos con un atributo llamado «serialNumber» que tiene un valor dentro del rango de cadenas alfanuméricas. Esta consulta devuelve

Cadena de consulta	Resultado
	objetos con un atributo "serialNumber" con valores "abcd", "abce" o "abcf".
attributes.myAttribute:i*t	Consulta los objetos con un atributo llamado «MyAttribute» donde el valor es 'i', seguido de un número de caracteres, seguido por 't'.
attributes.attr1:abc AND attributes.attr2<5 NOT attributes.attr3>10	Consultas de objetos que combinan términos mediante expresiones booleanas. Esta consulta devuelve objetos que tengan un atributo llamado "attr1" con un valor "abc", un atributo denominado "attr2" inferior a 5 y un atributo llamado "attr3" que no sea superior a 10.
shadow.hasDelta:true	Consulta los objetos con una sombra sin nombre que tenga un elemento delta.
NOT attributes.model:legacy	Consultas de objetos donde el atributo llamado "model" no es "legacy".
shadow.reported.stats.battery:{70 TO 100} (v2 OR v3) NOT attributes.model:legacy	Consulta los objetos que cumplen lo siguiente: <ul style="list-style-type: none"> El atributo stats.battery de sombra del objeto tiene un valor entre 70 y 100. El texto "v2" o "v3" aparece en los valores del atributo, el nombre del tipo o el nombre del objeto. El atributo model del objeto no está establecido en "legacy".
shadow.reported.myvalues:2	Consulta los objetos cuya matriz myvalues de la sección reported de la sombra contiene el valor 2.
shadow.reported.location:* NOT shadow.desired.stats.battery:*	Consulta los objetos que cumplen lo siguiente: <ul style="list-style-type: none"> El atributo location existe en la sección reported de la sombra. La stats.batteryatributo no existe en la sombra desiredsección.
shadow.name. <shadowName>. Hasdelta: Verdadero	Consultas para cosas que tienen una sombra con el nombre de pila y también un elemento delta. La función de indexación de flotas para admitir la indexación denominada shadow se encuentra en la versión preliminar de AWS IoT Administración de dispositivos.
shadow.name. <shadowName>.deseado.filamento: *	Consultas para cosas que tienen una sombra con el nombre de pila y también la propiedad de filamento deseada. La función de indexación de flotas para admitir la indexación denominada shadows está en versión preliminar de AWS IoT Administración de dispositivos.

Cadena de consulta	Resultado
shadow.name. <shadowName>.reported.location: *	Consultas para cosas que tienen una sombra con el nombre de pila y donde el location existe en la sección informada de la sombra con nombre. La función de indexación de flotas para admitir la indexación denominada shadows está en versión preliminar de AWS IoT Administración de dispositivos.
connectivity.connected:true	Consulta sobre todos los dispositivos conectados.
connectivity.connected:false	Consulta de todos los dispositivos desconectados.
connectivity.connected:true AND connectivity.timestamp: [1557651600000 A 1557867600000]	Consulta de todos los dispositivos conectados con una marca temporal de conexión >= 1557651600000 y <= 1557867600000. Las marcas temporales se indican en milisegundos desde la fecha de inicio.
connectivity.connected:false AND connectivity.timestamp: [1557651600000 A 1557867600000]	Consulta de todos los dispositivos desconectados con una marca temporal de desconexión >= 1557651600000 y <= 1557867600000. Las marcas temporales se indican en milisegundos desde la fecha de inicio.
connectivity.connected:true AND connectivity.timestamp > 1557651600000	Consulta de todos los dispositivos conectados con una marca temporal de conexión > 1557651600000. Las marcas temporales se indican en milisegundos desde la fecha de inicio.
connectivity.connected:*	Consulta todos los dispositivos para los que hay información de conectividad.
Conectividad.Desconexión Razón: *	Las consultas de todos los dispositivos con conectividad DisconnectReason están presentes.
Conectividad.DisconnectRazón: client_initiated_disconnect	Consultas para todos los dispositivos desconectados debido a CLIENT_INITIATED_DISCONNECT.
DeviceDefender.violationCount: [0 A 100]	Las consultas de objetos con infracciones de Device Defender cuentan el valor del rango numérico (entre 0 y 100, ambos incluidos). La función de indexación de flotas que admite la indexación de datos de infracciones de Device Defender está en versión preliminar AWS IoT Administración de dispositivos.
Defensor del dispositivo. <device-SecurityProfile>. Comportamiento de desconexión. Inviolación: Verdadero	Las consultas de objetos que violan el comportamiento disconnectBehavior según se define en el perfil de seguridad device-SecurityProfile. Tenga en cuenta que En violación: falso no es una consulta válida. La función de indexación de flotas que admite la indexación de datos de infracciones de Device Defender está en versión preliminar AWS IoT Administración de dispositivos.

Cadena de consulta	Resultado
Defensor del dispositivo. <device-SecurityProfile>.DisconnectBehavior.LastViolationValue.Numer>2	Las consultas de objetos que violan el comportamiento <code>disconnectBehavior</code> tal como se define en el perfil de seguridad <code>Device-SecurityProfile</code> con un valor de último evento de infracción superior a 2. La función de indexación de flotas que admite la indexación de datos de infracciones de Device Defender está en versión preliminar AWS IoT Administración de dispositivos.
Defensor del dispositivo. <device-SecurityProfile>.Disconnect Behavior.Tiempo de la última infracción > 1634227200000	Las consultas de objetos que violan el comportamiento <code>disconnectBehavior</code> tal como se define en el perfil de seguridad <code>Device-SecurityProfile</code> con un último evento de infracción tras un periodo determinado. La función de indexación de flotas que admite la indexación de datos de infracciones de Device Defender está en versión preliminar AWS IoT Administración de dispositivos.

Ejemplo de consultas de grupo de objetos

Las consultas se especifican en una cadena de consulta mediante una sintaxis de consulta y se trasladan a la API de [SearchIndex](#). En la siguiente tabla se enumeran algunas cadenas de consulta de ejemplo.

Cadena de consulta	Resultado
abc	Consulta "abc" en cualquier campo.
Nombres de grupo de cosas: mygroupThingName	Consulta un grupo de objetos con el nombre "myGroupThingName".
Nombres de grupos de cosas: my*	Consulta los grupos de objetos cuyos nombres que comienzan por "my".
Nombres de grupos de cosas: AB?	Consulta los grupos de objetos cuyos nombres tienen "ab" además de un carácter adicional (por ejemplo: "aba", "abb", "abc", etc.)
attributes.myAttribute:75	Consulta los grupos de con un atributo llamado "MyAttribute" que tiene el valor de 75.
attributes.myAttribute:[75 TO 80]	Consulta los grupos de con un atributo llamado «MyAttribute», cuyo valor se encuentra dentro del rango numérico (entre 75 y 80, ambos incluidos).
attributes.myAttribute:[75 TO 80]	Consulta los grupos de objetos con un atributo llamado "MyAttribute", cuyo valor se encuentra dentro del rango numérico (>75 y <=80).
attributes.myAttribute:["abcd" TO "abcf"]	Consulta los grupos de con un atributo llamado "myAttribute", cuyo valor se encuentra dentro del rango de cadenas alfanuméricas. Esta consulta devuelve grupos de objetos con un atributo "serialNumber" con valores "abcd", "abce" o "abcf".

Cadena de consulta	Resultado
attributes.myAttribute:i*t	Consulta los grupos de objetos con un atributo llamado "MyAttribute" cuyo valor es 'i', seguido de un número de caracteres, seguido por 't'.
attributes.attr1:abc AND attributes.attr2<5 NOT attributes.attr3>10	Consultas de grupos de objetos que combinan términos mediante expresiones booleanas. Esta consulta devuelve grupos de objetos que tengan un atributo denominado "attr1" con un valor "abc", un atributo denominado "attr2" inferior a 5 y un atributo denominado "attr3" que no sea superior a 10.
NOT attributes.myAttribute:cde	Consulta los grupos de objetos donde el atributo llamado "MyAttribute" no es "cde".
parentGroupNames:(myParentThingGroupName)	Consulta los grupos de objetos cuyo nombre de grupo principal coincide con "myParentThingGroupName".
parentGroupNames:(myParentThingGroupName OR myRootThingGroupName)	Consulta los grupos de objetos cuyo nombre de grupo principal coincide con "myParentThingGroupName" o "myRootThingGroupName".
parentGroupNames:(myParentThingGroupNa*)	Consulta los grupos de objetos cuyo nombre de grupo principal empieza por "myParentThingGroupNa".

Métricas de flota

La métrica de flota es una característica de [indexación de flota \(p. 825\)](#), un servicio administrado que permite indexar, realizar búsquedas y agregar los datos de sus dispositivos en AWS IoT. Con las métricas de flota, puede supervisar el estado agregado de sus dispositivos de flota en [CloudWatch](#) a lo largo del tiempo, incluida la revisión de la tasa de desconexión de los dispositivos de su flota o los cambios medios del nivel de batería de un período determinado.

Con las métricas de flota, puedes crear [consultas de agregación \(p. 840\)](#) cuyos resultados se emiten continuamente a [CloudWatch](#) como métricas para analizar tendencias y crear alarmas. Para las tareas de supervisión, puede especificar las consultas de agregación de los distintos tipos de agregación (Estadísticas, Cardinalidad, y Percentil). Puede guardar todas las consultas de agregación para crear métricas de flota para reutilizarlas en el futuro.

Explicación introductoria

En este tutorial, va a crear una [Representación métrica \(p. 851\)](#) para supervisar la temperatura de los sensores para detectar posibles anomalías. Al crear la métrica de flota, se define una [consulta de agregación \(p. 840\)](#) que detecta el número de sensores con temperaturas superiores a 80 grados Fahrenheit. Especifica la consulta que se ejecutará cada 60 segundos y los resultados de la consulta se emiten a CloudWatch, donde puede ver el número de sensores que tienen riesgos potenciales para altas temperaturas y configurar alarmas. Para completar este tutorial, utilizará [AWS CLI](#).

En este tutorial, aprenderá a:

- [Configurar \(p. 852\)](#)

- Creación de métricas de flota (p. 853)
- Ver métricas de CloudWatch (p. 855)
- Limpiar recursos (p. 856)

Este tutorial tarda unos 15 minutos en completarse.

Requisitos previos

- Instale la versión más reciente de [AWS CLI](#)
- Estar familiarizado con[Consulta de datos agregados](#)
- Estar familiarizado con[Uso de métricas de Amazon CloudWatch](#)

Configurar

Para utilizar las métricas de flota, debe habilitar la indexación de flotas. Para habilitar la indexación de flotas para sus elementos o grupos de cosas con orígenes de datos especificados y configuraciones asociadas, siga las instrucciones de[Administración de la indexación de objetos \(p. 829\)](#)y[Administración de la indexación de grupos de objetos \(p. 839\)](#).

Para configurar

1. Ejecute el siguiente comando para habilitar la indexación de flotas y especificar los orígenes de datos desde los que buscar.

```
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" \
```

El comando de CLI de ejemplo anterior permite la indexación de flota para permitir la búsqueda de datos de registro, datos de sombra y estado de conectividad del objeto mediante el AWS_Things índice.

El cambio de configuración puede tardar varios minutos en completarse. Asegúrese de que la indexación de flota esté habilitada antes de crear métricas de flota.

Para comprobar si la indexación de flota está habilitada, ejecute el siguiente comando de la CLI:

```
aws --region us-east-1 iot describe-index --index-name "AWS_Things"
```

Para obtener más información, consulte[Habilitar la indexación \(p. 829\)](#).

2. Ejecute el siguiente script bash para crear diez cosas y describirlas.

```
# 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 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
```

Este script crea diez elementos para representar diez sensores. Cada cosa tiene atributos `temperature`, `rackId`, y `stateNormal` tal como se describe en el cuadro siguiente:

Atributo	Tipo de datos	Descripción
<code>temperature</code>	Número	Valor de temperatura en Fahrenheit
<code>rackId</code>	Cadena	ID del rack de servidores que contiene sensores
<code>stateNormal</code>	Booleano	Tiempo en que el valor de temperatura del sensor es normal o no

La salida de este script contiene diez archivos JSON. Uno de los archivos JSON tiene este aspecto:

```
{  
    "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 obtener más información, consulte [Crear objeto](#).

Creación de métricas de flota

Para crear una métrica de flota.

1. Ejecute el siguiente comando para crear una métrica de flota denominada `High_temp_fm`. Cree la métrica de flota para supervisar el número de sensores con temperaturas superiores a 80 grados Fahrenheit en 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
```

—nombre métrico

Tipo de datos: cadena. La `--metric-name` especifica un nombre de métrica de flota. En este ejemplo, va a crear una métrica de flota denominada `High_temp_fm`.

—cadena de consulta

Tipo de datos: cadena. La `--query-string` parámetro especifica la cadena de consulta. En este ejemplo, la cadena de consulta significa consultar todas las cosas con nombres que empiezan

porTempSensory con temperaturas superiores a 80 grados Fahrenheit. Para obtener más información, consulte[Sintaxis de la consulta \(p. 846\)](#).

—punto

Tipo de datos: número entero. La--periodespecifica el tiempo de recuperación de los datos agregados en segundos. En este ejemplo, especifica que la métrica de flota que está creando recupera los datos agregados cada 60 segundos.

—campo de agregación

Tipo de datos: cadena. La--aggregation-fieldespecifica el atributo que se va a evaluar. En este ejemplo, se va a evaluar el atributo de temperatura.

—tipo agregación

La--aggregation-typeespecifica el resumen estadístico que se mostrará en la métrica de flota. Para las tareas de supervisión, puede personalizar las propiedades de la consulta de agregación para los distintos tipos de agregación (Estadísticas, Cardinalidad, yPercentil). En este ejemplo, debe especificar conteo para el tipo de agregación Estadísticas para devolver el número de dispositivos que tienen atributos que coinciden con la consulta; en otras palabras, para devolver el recuento de dispositivos con nombres que empiecen porTempSensory con temperaturas superiores a 80 grados Fahrenheit. Para obtener más información, consulte[Consulta de datos agregados \(p. 840\)](#).

El resultado de este comando tendrá un aspecto similar al siguiente.

```
{  
    "metricArn": "arn:aws:iot:region:111122223333:fleetmetric/high_temp_FM",  
    "metricName": "high_temp_FM"  
}
```

Note

Los puntos de datos pueden tardar un momento en mostrarse en CloudWatch.

Para obtener más información acerca de cómo crear una métrica de flota, lea[Administración de métricas de flota \(p. 856\)](#).

Si no puedes crear una métrica de flota, lee[Solución de problemas con métricas de flota \(p. 1285\)](#).

2. (Opcional) Ejecute el siguiente comando para describir la métrica de flota denominadaHigh_temp_fm:

```
aws iot describe-fleet-metric --metric-name "high_temp_FM"
```

El resultado de este comando tendrá un aspecto similar al siguiente.

```
{  
    "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"
}
```

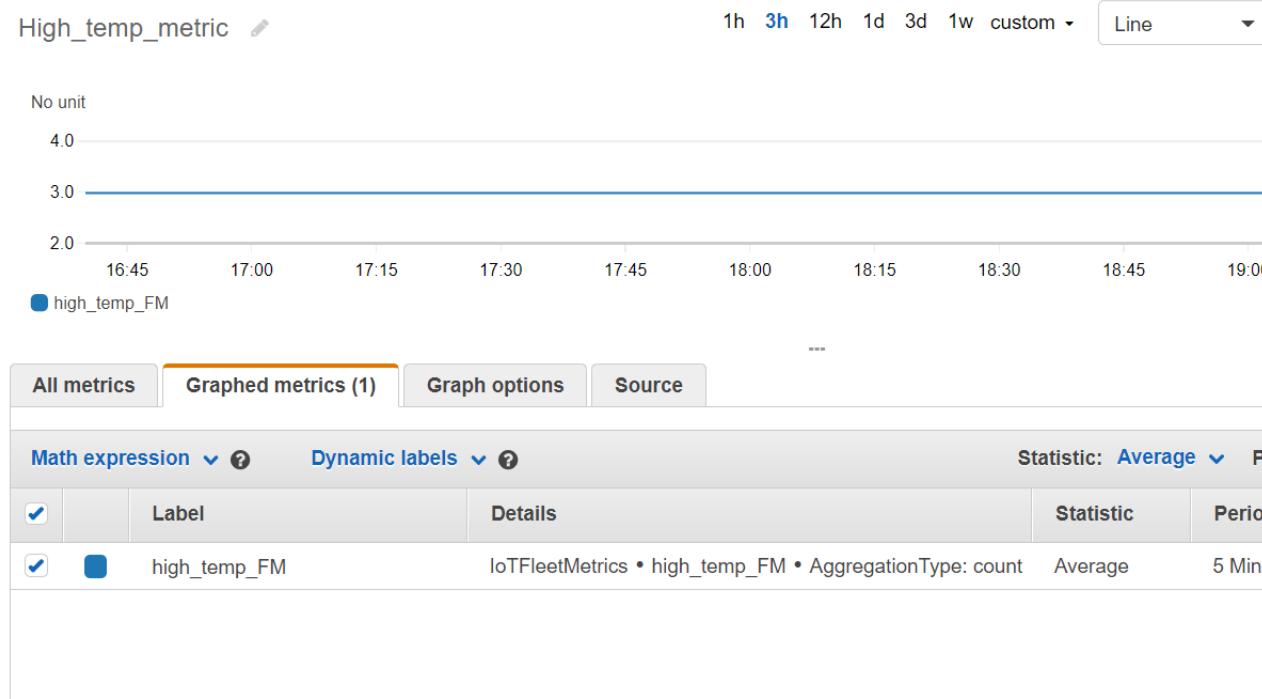
Ver métricas de flota en CloudWatch

Después de crear la métrica de flota, puede ver los datos de métricas en CloudWatch. En este tutorial, verás la métrica que muestra el número de sensores con nombres que empiezan por TempSensory con temperaturas superiores a 80 grados Fahrenheit.

Para ver los puntos de datos en CloudWatch

1. Abra la consola de CloudWatch en <https://console.aws.amazon.com/cloudwatch/>.
2. En el menú de CloudWatch de la izquierda, elija **Métricas** para ampliar el submenú y, a continuación, elija **Todas las métricas**. Se abre la página con la mitad superior para mostrar el gráfico y la mitad inferior que contiene cuatro secciones con pestañas.
3. La primera sección con pestañas **Todas las métricas** enumera todas las métricas que puedes ver en grupos, elige **Métricas de flota** que contiene todas las métricas de tu flota.
4. En la página **Tipo de agregación** sección sobre de la **Todas las métricas** pestaña, elija **Tipo de agregación** para ver todas las métricas de flota que has creado.
5. Elija la métrica de flota para mostrar el gráfico a la izquierda del **Tipo de agregación** sección. Verás el **valor suma** la izquierda de su **Nombre de métrica**, y este es el valor del tipo de agregación que especificó en el [Creación de métricas de flota \(p. 853\)](#) sección de este tutorial.
6. Elija la segunda pestaña denominada **Representación gráfica** de métricas a la derecha de la **Todas las métricas** para ver la métrica de flota que eligió del paso anterior.

Debería poder ver un gráfico que muestre el número de sensores con temperaturas superiores a 80 grados Fahrenheit como se indica a continuación:



Note

La `PeriodoEl` atributo de CloudWatch tiene un valor predeterminado de 5 minutos. Es el intervalo de tiempo entre los puntos de datos que se muestran en CloudWatch. Puede cambiar el `Periodo` en función de sus necesidades.

7. (Opcional) Puede configurar una alarma métrica.

1. En el menú de CloudWatch de la izquierda, elija `Alarms` para ampliar el submenú y, a continuación, elija `Todas las alarmas`.
2. En la página `Alarms`, elija `Crear alarma` en la esquina superior derecha. Siga las instrucciones en la consola para crear una alarma según sea necesario. Para obtener más información, consulte [Uso de alarmas de Amazon CloudWatch](#).

Para obtener más información, lea [Uso de métricas de Amazon CloudWatch](#).

Si no puedes ver los puntos de datos en CloudWatch, lee [Solución de problemas con métricas de flota \(p. 1285\)](#).

Eliminar recursos

Para eliminar métricas de flota

Usa el `delete-fleet-metric` comando de la CLI para eliminar métricas de flota.

Ejecute el siguiente comando para eliminar la métrica de flota denominada `High_temp_fm`.

```
aws iot delete-fleet-metric --metric-name "high_temp_FM"
```

Para limpiar las cosas

Usa el `delete-thing` comando de la CLI para eliminar objetos.

Ejecute el siguiente script para eliminar las diez cosas que creó:

```
# 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 limpiar las métricas de CloudWatch

CloudWatch no admite la eliminación de métricas. Las métricas caducan según sus cronogramas de retención. Para obtener más información, consulte [Uso de métricas de Amazon CloudWatch](#).

Administración de métricas de flota

En este tema se muestra cómo utilizar la AWS IoT Consola y la AWS CLI para administrar las métricas de su flota.

Administración de métricas de flota (consola)

Asegúrese de haber habilitado la indexación de flotas con fuentes de datos y configuraciones asociadas antes de crear métricas de flota.

Habilitación de indexación

Si ya has habilitado la indexación de flotas, omite esta sección.

Si no ha habilitado la indexación de flota, siga estas instrucciones.

1. Abra suAWS IoTConsola de en<https://console.aws.amazon.com/iot/>.
2. En la páginaAWS IoTmenú, elijaConfiguración.
3. Para ver la configuración detallada, en elConfiguración, vaya a la páginaindexación de flotassección.
4. Para actualizar la configuración de indexación de flotas, a la derecha dellIndexación de flotassección, seleccioneAdministrar la indexación.
5. En la páginaGestionar la indexación, actualice la configuración de indexación de flota en función de sus necesidades.
 - Configuración

Para activar la indexación de cosas, alternarIndexación de objetosy, a continuación, seleccione los orígenes de datos desde los que desea indexar.

Para activar la indexación de grupos de cosas, alternarIndexación de grupos de objetosen.

- Campos de búsqueda personalizados -opcional

Los campos de búsqueda personalizados son una lista de pares de nombres de campo y tipos de campo.

Para añadir un par de campos personalizado, elijaAregar nuevo campo. Escriba un nombre de campo personalizado, como`attributes.temperature`y, a continuación, seleccione un tipo de campo de laTipo de campomenú. Tenga en cuenta que el nombre de un campo personalizado comienza por`attributes.`y se guardará como atributo para ejecutar[Consultas de agregaciones de objetos](#).

Para actualizar y guardar la configuración, elijaActualización.

Crear una métrica de flota

1. Abra suAWS IoTConsola de en<https://console.aws.amazon.com/iot/>.
2. En la páginaAWS IoTmenú, elijaManejary luego elijaMétricas de flota.
3. En la páginaMétricas de flotapágina, elijaCrear métrica de flotay completa los pasos de creación.
4. En el paso 1Configurar métricas de flota
 - EnConsulta, introduzca una cadena de consulta para especificar las cosas o grupos de cosas que desea realizar la búsqueda agregada. La cadena de consulta consta de un atributo y un valor. ParaPropiedades, elija el atributo que deseé o, si no aparece en la lista, introduzca el atributo en el campo. Introduzca el valor después de:. Una cadena de consulta de ejemplo puede ser`thingName : TempSensor*`. Para cada cadena de consulta que introduzcas, presionaentraren el teclado. Si introduce varias cadenas de consulta, especifique su relación seleccionandoy,o,y no, o bieno nentre ellos.
 - EnPropiedades de informe, eligeNombre de índice,Tipo de agregación, yCampo de agregaciónde sus respectivas listas. A continuación, seleccione los datos que desea agregar enSeleccionar datos, donde puede seleccionar varios valores de datos.
 - Elija Next (Siguiente).
5. En el paso 2Especificar las propiedades de métricas
 - EnNombre métrica de flota, escriba un nombre para la métrica de flota que está creando.
 - EnDescripción -opcional, escriba una descripción de la métrica de flota que está creando. Este campo es opcional.

- En HoursyActa, introduzca la hora (con qué frecuencia) que desea que la métrica de flota emita datos a CloudWatch.
 - Elija Next (Siguiente).
6. En el paso 3Revisar y crear
- Revise la configuración de los pasos 1 y 2. Para editar esta configuración, elija Editar.
 - ElegirCrear métrica de flota.

Tras la creación satisfactoria, la métrica de flota aparece en elMétricas flotas(Se ha creado el certificado).

Actualización de una métrica de flota

1. En la páginaMétricas flotas, elija la métrica de flota que desea actualizar.
2. En la métrica de flotaDetalles depágina, elija Editar. Esto abre los pasos de creación en los que puedes actualizar la métrica de tu flota en cualquiera de los tres pasos.
3. Una vez que haya terminado de actualizar la métrica de flota, elija Actualización de la métrica de.

Eliminar una métrica de flota

1. En la páginaMétricas flotas, elija la métrica de flota que desea eliminar.
2. En la página siguiente que muestra los detalles de la métrica de su flota, elija Borrar.
3. En el cuadro de diálogo, escriba el nombre de la métrica de flota para confirmar la eliminación.
4. Elija Delete (Eliminar). En este paso se elimina la métrica de flota de forma permanente.

Administración de métricas de flota (CLI)

En las siguientes secciones se muestra cómo utilizar elAWSCLI para administrar las métricas de su flota. Asegúrese de haber habilitado la indexación de flotas con fuentes de datos y configuraciones asociadas antes de crear métricas de flota. Para habilitar la indexación de flota para sus objetos o grupos de objetos, siga las instrucciones de[Administración de la indexación de objetos \(p. 829\)](#)o[Administración de la indexación de grupos de objetos \(p. 839\)](#).

Crear una métrica de flota

Utilice el comando create-fleet-metric CLI para crear una métrica de flota.

```
aws iot create-fleet-metric --metric-name "YourFleetMetricName" --query-string "*" --period 60 --aggregation-field "registry.version" --aggregation-type name=Statistics,values=sum
```

La salida de este comando contiene el nombre y el nombre de recurso de Amazon (ARN) de la métrica de flota. El resultado es similar al siguiente:

```
{  
    "metricArn": "arn:aws:iot:us-east-1:111122223333:fleetmetric/YourFleetMetricName",  
    "metricName": "YourFleetMetricName"  
}
```

Listar métricas de flota

Utilice el comando list-fleet-metric CLI para enumerar todas las métricas de flota de su cuenta.

```
aws iot list-fleet-metrics
```

La salida de este comando contiene todas las métricas de flota. El resultado es similar al siguiente:

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

Describa una métrica de

Utilice el comando describe-fleet-metric CLI para mostrar información más detallada sobre una métrica de flota.

```
aws iot describe-fleet-metric --metric-name "YourFleetMetricName"
```

La salida del comando contiene la información detallada acerca de la métrica de flota especificada. El resultado es similar al siguiente:

```
{  
    "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"  
}
```

Actualización de una métrica de flota

Utilice el comando de CLI update-fleet-metric para actualizar una métrica de flota.

```
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
```

El comando update-fleet-metric no produce ningún resultado. Puede utilizar el comando describe-fleet-metric de la CLI para ver el 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"  
}
```

Eliminar una métrica de flota

Utilice el comando `delete-fleet-metric` CLI para eliminar una métrica de flota.

```
aws iot delete-fleet-metric --metric-name "YourFleetMetricName"
```

Este comando no produce ningún resultado si la eliminación se realiza correctamente o especifica una métrica de flota que no existe.

Para obtener más información, consulte [Solución de problemas con métricas de flota \(p. 1285\)](#).

Entrega de archivos basada en MQTT

Una opción que puedes usar para administrar archivos y transferirlos aAWS IoT los dispositivos de su flota son la entrega de archivos basada en MQTT. Con esta característica enAWSCloud puedes crear unflujo deque contiene varios archivos, puede actualizar los datos de transmisión (la lista de archivos y las descripciones), obtener los datos de transmisión y mucho más.AWS IoT La entrega de archivos basada en MQTT puede transferir datos en bloques pequeños a sus dispositivos IoT, utilizando el protocolo MQTT compatible con mensajes de solicitud y respuesta en JSON o CBOR.

Para obtener más información sobre las formas de transferir datos desde y hacia dispositivos IoT medianteAWS IoT, consulte[Conexión de dispositivos aAWS IoT \(p. 78\)](#).

Temas

- [¿Qué es un flujo? \(p. 861\)](#)
- [Administración de una transmisión en elAWSNube \(p. 862\)](#)
- [Uso deAWS IoTEntrega de archivos basada en MQTT en dispositivos \(p. 863\)](#)
- [Ejemplo de caso de uso en FreeRTOS OTA \(p. 870\)](#)

¿Qué es un flujo?

EnAWS IoT, unflujo dees un recurso direccionable públicamente que es una abstracción de una lista de archivos que se pueden transferir a un dispositivo IoT. Un flujo típico contiene la siguiente información:

- Un nombre de recurso de Amazon (ARN)que identifica de forma única un flujo en un momento dado. Este ARN tiene el patrón`narn:partition:iot:region:account-ID:stream/stream ID`.
- Un ID de flujoque identifica la transmisión y se utiliza (y normalmente es necesario) enAWS Command Line Interface(AWS CLI) o comandos SDK.
- Descripción de una transmisiónque proporciona una descripción del recurso de secuencia.
- Una versión de streamingque identifica una versión determinada de la secuencia. Dado que los datos de transmisión se pueden modificar inmediatamente antes de que los dispositivos inicien la transferencia de datos, los dispositivos pueden utilizar la versión de transmisión para hacer cumplir una comprobación de coherencia.
- Una lista de archivosque se pueden transferir a dispositivos. Para cada archivo de la lista, el flujo registra un ID de archivo, el tamaño de archivo y la información de dirección del archivo, que consiste, por ejemplo, en el nombre del depósito de Amazon S3, la clave de objeto y la versión del objeto.
- UnAWS Identity and Access Management(IAM)que concedeAWS IoTEntrega de archivos basada en MQTT el permiso para leer los archivos de transmisión almacenados en el almacenamiento de datos.

AWS IoTLa entrega de archivos basada en MQTT proporciona la siguiente funcionalidad para que los dispositivos puedan transferir datos desde elAWSCloud:

- Transferencia de datos mediante el protocolo MQTT.
- Support formatos JSON o CBOR.
- La capacidad de describir un flujo ([DescribeStreamAPI](#)) para obtener una lista de archivos de transmisión, una versión de transmisión e información relacionada.
- Posibilidad de enviar datos en bloques pequeños ([GetStreamAPI](#)) para que los dispositivos con restricciones de hardware puedan recibir los bloques.

- Soporta un tamaño de bloque dinámico por solicitud, para admitir dispositivos que tienen capacidades de memoria diferentes.
- Optimización para solicitudes de streaming simultáneas cuando varios dispositivos solicitan bloques de datos del mismo archivo de transmisión.
- Amazon S3 como almacenamiento de datos para archivos de transmisión.
- Soporta la publicación de registros de transferencia de datos desde AWS IoT a CloudWatch.

Para obtener información sobre las cuotas de entrega de archivos basadas en MQTT, consulte [AWS IoT Core Cuotas de servicio](#) en la AWS Referencia general de.

Administración de una transmisión en elAWSNube

AWS IoT proporciona AWSSDK y AWS CLI comandos que puede utilizar para administrar una secuencia en la AWS Cloud. Puede utilizar estos comandos para hacer lo siguiente:

- Crea un flujo. [CLI/SDK](#)
- Describa una transmisión para obtener su información. [CLI/SDK](#)
- Mostrar flujos en su cuenta de AWS. [CLI/SDK](#)
- Actualiza la lista de archivos o la descripción de la transmisión de una transmisión. [CLI/SDK](#)
- Elimina un flujo. [CLI/SDK](#)

Note

En este momento, las transmisiones no son visibles en el AWS Management Console. Debe utilizar el AWS CLI o AWSSDK para administrar una transmisión en AWS IoT.

Antes de usar AWS IoT para entregar archivos basados en MQTT desde tus dispositivos, debes seguir los pasos de las siguientes secciones para asegurarte de que tus dispositivos estén debidamente autorizados y se puedan conectar al AWS IoT Gateway para dispositivos.

Conceder permisos a tus dispositivos

Puede seguir los pasos en [Creación de una política de dispositivo](#) para crear una política de dispositivo o utilizar una política de dispositivo existente. Adjunte la política a los certificados asociados a los dispositivos y agregue los siguientes permisos a la directiva de dispositivos.

```
{  
    "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 los dispositivos aAWS IoT

Dispositivos que utilizan AWS IoT requieren la entrega de archivos basada en MQTT para conectarse con AWS IoT. La entrega de archivos basada en MQTT se integra con AWS IoT en la AWS Cloud, por lo que tus dispositivos deben conectarse directamente al punto final de la AWS IoT Plan de datos de.

Note

El punto de enlace de la AWS IoT Plan de datos es específico de la cuenta de AWS y la región. Debe utilizar el endpoint para la cuenta de AWS y la región en la que están registrados los dispositivos AWS IoT.

Para obtener más información, consulte [Conexión a AWS IoT Core \(p. 69\)](#).

Uso de AWS IoT Entrega de archivos basada en MQTT en dispositivos

Para iniciar el proceso de transferencia de datos, un dispositivo debe recibir un conjunto de datos iniciales, que incluye un identificador de flujo como mínimo. Puede usar un [Trabajos \(p. 675\)](#) para programar tareas de transferencia de datos para sus dispositivos mediante la inclusión del conjunto de datos inicial en el documento de trabajo. Cuando un dispositivo recibe el conjunto de datos inicial, debe iniciar la interacción con AWS IoT Entrega de archivos basada en MQTT. Para intercambiar datos con AWS IoT Entrega de archivos basada en MQTT, un dispositivo debe:

- Utilice el protocolo MQTT para suscribirse al [Temas de entrega de archivos basados en MQTT \(p. 115\)](#).
- Envíe solicitudes y, a continuación, espera a recibir las respuestas mediante mensajes MQTT.

Opcionalmente, puede incluir un ID de archivo de transmisión y una versión de transmisión en el conjunto de datos inicial. El envío de un ID de archivo de transmisión a un dispositivo puede simplificar la programación del firmware/software del dispositivo, ya que elimina la necesidad de crear una `DescribeStream` solicitud desde el dispositivo para obtener este ID. El dispositivo puede especificar la versión de transmisión en una `GetStream` solicitud para hacer cumplir una comprobación de coherencia en caso de que la transmisión se haya actualizado inesperadamente.

Utilizar `DescribeStream` para obtener datos de transmisión

AWS IoT La entrega de archivos basada en MQTT proporciona la `DescribeStream` API para enviar datos de transmisión a un dispositivo. Los datos de transmisión devueltos por esta API incluyen el ID de transmisión, la versión de transmisión, la descripción de la transmisión y una lista de archivos de transmisión, cada uno de los cuales tiene un ID de archivo y el tamaño del archivo en bytes. Con esta

información, un dispositivo puede seleccionar archivos arbitrarios para iniciar el proceso de transferencia de datos.

Note

No tiene que usar el `DescribeStream` API si el dispositivo recibe todos los ID de archivo de flujo requeridos en el conjunto de datos inicial.

Siga estos pasos para crear `DescribeStream` request.

1. Suscribirse al filtro de temas «aceptados»`$aws/things/ThingName/streams/StreamId/description/json`.
2. Suscribirse al filtro de temas «rechazados»`$aws/things/ThingName/streams/StreamId/rejected/json`.
3. Publicación de una `DescribeStream` solicitud enviando un mensaje a`$aws/things/ThingName/streams/StreamId/describe/json`.
4. Si se acepta la solicitud, el dispositivo recibe una `DescribeStream` respuesta en el filtro de temas «aceptados».
5. Si se ha rechazado la solicitud, el dispositivo recibe la respuesta de error en el filtro de temas «rechazados».

Note

Si reemplaza `json` con `cbor` en los filtros de temas y temas que se muestran, el dispositivo recibe mensajes en formato CBOR, que es más compacto que JSON.

DescribeStream request

Un típico `DescribeStream` solicitud en JSON es similar al del siguiente ejemplo.

```
{  
    "c": "ec944cfb-1e3c-49ac-97de-9dc4aaad0039"  
}
```

- (Opcional) «c» es el campo token de cliente.

El token de cliente no puede tener más de 64 bytes. Un token de cliente que es superior a 64 bytes provocará una respuesta de error y un `InvalidRequest`.

DescribeStream response

Una `DescribeStream` respuesta en JSON es similar al del siguiente ejemplo.

```
{  
    "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«es el campo token de cliente. Esto se devuelve si se ha dado en elDescribeStreamrequest. Utilice el token de cliente para asociar la respuesta a su solicitud.
- "s«es la versión de transmisión como un entero. Puede utilizar esta versión para realizar una comprobación de coherencia con suGetStreamsolicitudes.
- "r«contiene una lista de archivos de la secuencia.
 - "f«es el ID del archivo de transmisión como entero.
 - "z«es el tamaño del archivo de transmisión en número de bytes.
- "d«contiene la descripción del flujo.

Obtener bloques de datos de un archivo de transmisión

Puede utilizar elGetStreamAPI para que un dispositivo pueda recibir archivos de transmisión en pequeños bloques de datos, por lo que los dispositivos que tienen restricciones para procesar bloques de gran tamaño. Para recibir un archivo de datos completo, es posible que un dispositivo tenga que enviar o recibir varias solicitudes y respuestas hasta que se reciban y procesen todos los bloques de datos.

request getStream

Siga estos pasos para crearGetStreamrequest.

1. Suscribirse al filtro de temas «aceptados»\$aws/things/*ThingName*/streams/*StreamId*/data/json.
2. Suscribirse al filtro de temas «rechazados»\$aws/things/*ThingName*/streams/*StreamId*/rejected/json.
3. Publicación de unGetStreamsolicitud al tema\$aws/things/*ThingName*/streams/*StreamId*/get/json.
4. Si se acepta la solicitud, el dispositivo recibirá uno o másGetStreamrespuestas en el filtro de temas «aceptados». Cada mensaje de respuesta contiene información básica y una carga útil de datos para un solo bloque.
5. Repita los pasos 3 y 4 para recibir todos los bloques de datos. Debe repetir estos pasos si la cantidad de datos solicitados supera los 128 KB. Debe programar el dispositivo para que use variosGetStreamsolicitudes para recibir todos los datos solicitados.
6. Si se ha rechazado la solicitud, el dispositivo recibirá la respuesta de error en el filtro de temas «rechazados».

Note

- Si sustituye «json» por «cbor» en los filtros de temas y temas mostrados, el dispositivo recibirá mensajes en formato CBOR, que es más compacto que JSON.
- AWS IoTLa entrega de archivos basada en MQTT limita el tamaño de un bloque a 128 KB. Si realiza una solicitud para un bloque de más de 128 KB, la solicitud fallará.
- Puede realizar una solicitud de varios bloques cuyo tamaño total sea superior a 128 KB (por ejemplo, si realiza una solicitud de 5 bloques de 32 KB cada uno para un total de 160 KB de datos). En este caso, la solicitud no falla, pero el dispositivo debe realizar varias solicitudes para recibir todos los datos solicitados. El servicio enviará bloques adicionales a medida que el dispositivo realice solicitudes adicionales. Te recomendamos que continúes con una nueva solicitud solo después de que la respuesta anterior se haya recibido y procesado correctamente.

- Independientemente del tamaño total de los datos solicitados, debe programar el dispositivo para iniciar reintentos cuando los bloques no se reciben o no se reciben correctamente.

Un típicoGetStream solicitud en JSON es similar al del siguiente ejemplo.

```
{  
    "c": "1bb8aaa1-5c18-4d21-80c2-0b44fee10380",  
    "s": 1,  
    "f": 0,  
    "l": 4096,  
    "o": 2,  
    "n": 100,  
    "b": "..."  
}
```

- [opcional]»c«es el campo token de cliente.

El token de cliente no puede ser superior a 64 bytes. Un token de cliente que es superior a 64 bytes provocará una respuesta de error y unInvalidRequest.

- [opcional]»s«es el campo de versión de transmisión (un entero).

La entrega de archivos basada en MQTT aplica una comprobación de coherencia basada en esta versión solicitada y en la última versión de transmisión en la nube. Si la versión de transmisión enviada desde un dispositivo en unGetStreamrequest no coincide con la última versión de transmisión en la nube, el servicio envía una respuesta de error y unVersionMismatch. Normalmente, un dispositivo recibe la versión de transmisión esperada (más reciente) en el conjunto de datos inicial o en la respuesta aDescribeStream.

- "f«es el ID del archivo de transmisión (un entero entre 0 y 255).

El ID del archivo de transmisión es obligatorio cuando se crea o actualiza una transmisión mediante elAWS CLIo SDK. Si un dispositivo solicita un archivo de transmisión con un ID que no existe, el servicio envía una respuesta de error y unResourceNotFound.

- "l«es el tamaño del bloque de datos en bytes (un entero entre 256 y 131.072).

Consulte[Crear un mapa de bits para una solicitud GetStream \(p. 867\)](#)para obtener instrucciones sobre cómo utilizar los campos de mapa de bits para especificar qué parte del archivo de transmisión se devolverá en elGetStreamresponse. Si un dispositivo especifica un tamaño de bloque que está fuera de rango, el servicio envía una respuesta de error y unBlockSizeOutOfBounds.

- [opcional]»o«es el desplazamiento del bloque del archivo de transmisión (un entero entre 0 y 98.304).

Consulte[Crear un mapa de bits para una solicitud GetStream \(p. 867\)](#)para obtener instrucciones sobre cómo utilizar los campos de mapa de bits para especificar qué parte del archivo de transmisión se devolverá en elGetStreamresponse. El valor máximo de 98.304 se basa en un límite de tamaño de archivo de flujo de 24 MB y 256 bytes para el tamaño de bloque mínimo. El valor predeterminado es 0 si no se especifica.

- [opcional]»n«es el número de bloques solicitados (un entero entre 0 y 98.304).

El campo »n« especifica (1) el número de bloques solicitados o (2) cuando se utiliza el campo de mapa de bits (»b«), un límite en el número de bloques que devolverá la solicitud de mapa de bits. Este segundo uso es opcional. Si no se define, el valor predeterminado es 131072/[Tamaño del bloque de datos](#).

- [opcional]»b«es un mapa de bits que representa los bloques que se están solicitando.

Con un mapa de bits, el dispositivo puede solicitar bloques no consecutivos, lo que hace que la gestión de los reintentos tras un error sea más conveniente. Consulte[Crear un mapa de bits para una solicitud GetStream \(p. 867\)](#)para obtener instrucciones sobre cómo utilizar los campos de mapa de bits para

especificar qué parte del archivo de transmisión se devolverá en elGetStreamresponse. Para este campo, convierta el mapa de bits en una cadena que representa el valor del mapa de bits en notación hexadecimal. El mapa de bits debe tener menos de 12.288 bytes.

Important

O sea»n»o»b«debe especificarse. Si no se especifica ninguno de ellos, elGetStreames posible que no sea válida si el tamaño del archivo es inferior a 131072 bytes (128 KB).

Respuesta GetStream

UNA GetStream respuesta en JSON se parece a este ejemplo para cada bloque de datos que se solicita.

```
{  
    "c": "1bb8aaa1-5c18-4d21-80c2-0b44fee10380",  
    "f": 0,  
    "l": 4096,  
    "i": 2,  
    "p": "..."  
}
```

- "c«es el campo token de cliente. Esto se devuelve si se ha dado en elGetStreamrequest. Utilice el token de cliente para asociar la respuesta a su solicitud.
- "f«es el ID del archivo de transmisión al que pertenece la carga útil del bloque de datos actual.
- "l«es el tamaño de la carga útil del bloque de datos en bytes.
- "i«es el ID del bloque de datos contenido en la carga útil. Los bloques de datos se enumeran a partir de 0.
- "p«contiene la carga útil del bloque de datos. Este campo es una cadena, que representa el valor del bloque de datos enBase64codificación.

Crear un mapa de bits para una solicitud GetStream

Puede utilizar el campo de mapa de bits (b) en unGetStreamsolicitud para obtener bloques no consecutivos de un archivo de transmisión. Esto ayuda a los dispositivos con capacidad RAM limitada a resolver problemas de entrega de red. Un dispositivo solo puede solicitar los bloques que no se recibieron o no se han recibido correctamente. El mapa de bits determina qué bloques del archivo de transmisión se devolverán. Para cada bit, que se establece en 1 en el mapa de bits, se devolverá un bloque correspondiente del archivo de transmisión.

A continuación, se muestra un ejemplo de cómo especificar un mapa de bits y sus campos de soporte en unGetStreamrequest. Por ejemplo, desea recibir un archivo de transmisión en fragmentos de 256 bytes (el tamaño de bloque). Piense en que cada bloque de 256 bytes tiene un número que especifica su posición en el archivo, empezando por 0. Así que el bloque 0 es el primer bloque de 256 bytes del archivo, el bloque 1 es el segundo y, así, sucesivamente. Desea solicitar los bloques 20, 21, 24 y 43 del archivo.

Bloque offset

Dado que el primer bloque es el número 20, especifique el desfase (campo)o) como 20 para ahorrar espacio en el mapa de bits.

Número de bloques

Para asegurarse de que el dispositivo no recibe más bloques de los que puede manejar con recursos de memoria limitados, puede especificar el número máximo de bloques que deben devolverse en cada mensaje enviado por la entrega de archivos basada en MQTT. Tenga en cuenta que este valor no se tiene en cuenta si el propio mapa de bits especifica menos que este número de bloques, o si haría que

el tamaño total de los mensajes de respuesta enviados por la entrega de archivos basada en MQTT sea superior al límite de servicio de 128 KB porGetStreamrequest.

Bloque bitmap

El mapa de bits en sí es una matriz de bytes sin firmar expresados en notación hexadecimal e incluidos en elGetStream solicitud como representación de cadena del número. Pero para construir esta cadena, empiezamos pensando en el mapa de bits como una larga secuencia de bits (un número binario). Si un poco de esta secuencia se establece en 1, el bloque correspondiente del archivo de transmisión se enviará de vuelta al dispositivo. Para nuestro ejemplo, queremos recibir los bloques 20, 21, 24 y 43, por lo que debemos establecer bits 20, 21, 24 y 43 en nuestro mapa de bits. Podemos utilizar el desplazamiento de bloque para ahorrar espacio, así que después de restar el desplazamiento de cada número de bloque, queremos establecer los bits 0, 1, 4 y 23, como en el siguiente ejemplo.

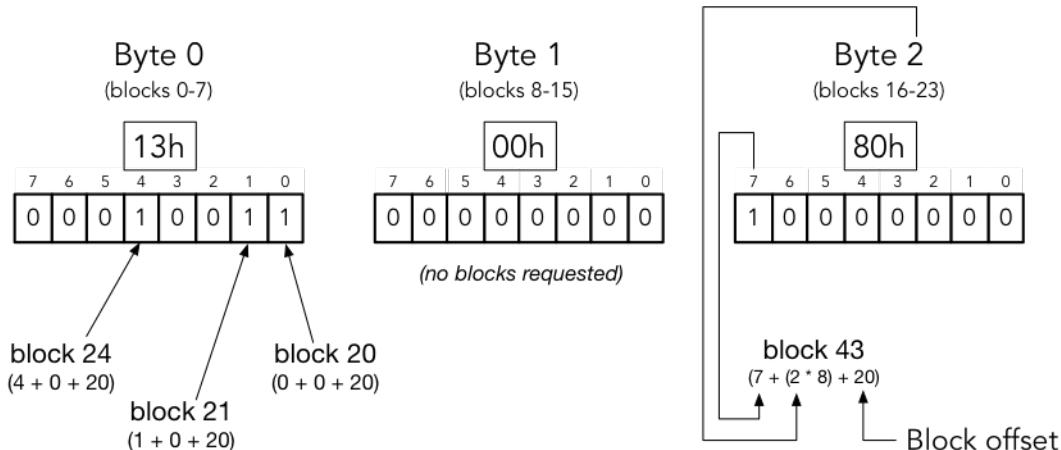
```
1 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
```

Tomando un byte (8 bits) a la vez, esto se escribe convencionalmente como: «0b00010011», «0b00000000» y «0b10000000». El bit 0 aparece en nuestra representación binaria al final del primer byte y el bit 23 al principio del último. Esto puede resultar confuso a menos que conozcas las convenciones. El primer byte contiene bits 7-0 (en ese orden), el segundo byte contiene bits 15-8, el tercer byte contiene bits 23-16, etc. En notación hexadecimal, esto se convierte en «0x130080».

Tip

Puede convertir el binario estándar en notación hexadecimal. Toma cuatro dígitos binarios a la vez y conviértelos a su equivalente hexadecimal. Por ejemplo, «0001» se convierte en «1», «0011» se convierte en «3» y así sucesivamente.

Block bitmap breakdown



$$\text{block number} = (\text{bit position} + (\text{byte offset} * 8) + \text{base offset})$$

Reuniendo todo esto, el JSON para nuestroGetStreamrequest tiene un aspecto similar al siguiente.

```
{
  "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
}
```

Gestión de errores de AWS IoTEntrega de archivos basada en MQTT

Respuesta de error que se envía a un dispositivo para ambos `DescribeStream` y `GetStream`. Las API contienen un token de cliente, un código de error y un mensaje de error. Una respuesta de error típica es similar al del siguiente ejemplo.

```
{
  "o": "BlockSizeOutOfBounds",
  "m": "The block size is out of bounds",
  "c": "1bb8aaa1-5c18-4d21-80c2-0b44fee10380"
}
```

- "`o`"«es el código de error que indica el motivo de un error. Consulte los códigos de error más adelante en esta sección para obtener más detalles.
- "`m`"«es el mensaje de error que contiene detalles del error.
- "`c`"«es el campo token de cliente. Esto se puede devolver si se ha dado en el `DescribeStream` request. Puede utilizar el token de cliente para asociar la respuesta a su solicitud.

El campo token de cliente no siempre se incluye en una respuesta de error. Cuando el token de cliente proporcionado en la solicitud no es válido o está mal formado, no se devuelve en la respuesta de error.

Note

Para la compatibilidad con versiones anteriores, los campos de la respuesta al error pueden estar en forma no abreviada. Por ejemplo, el código de error podría designarse mediante campos «código» o «o» y el campo de token de cliente puede designarse mediante campos «ClientToken» o «c». Le recomendamos que utilice el formulario de abreviatura que se muestra arriba.

InvalidTopic

El tema MQTT del mensaje de secuencia no es válido.

InvalidJson

La solicitud Stream no es un documento JSON válido.

CBOR inválida

La solicitud Stream no es un documento CBOR válido.

InvalidRequest

La solicitud se identifica generalmente como mal formada. Para obtener más información, consulte el mensaje de error.

Sin autorización

La solicitud no está autorizada para acceder a los archivos de datos de transmisión en el medio de almacenamiento, como Amazon S3. Para obtener más información, consulte el mensaje de error.

Tamaño de bloque fuera de límites

El tamaño del bloque está fuera de los límites. Consulte »Entrega de archivos basada en MQTT« sección en [AWS IoT Core](#) [Cuotas de servicio](#).

Desplazamiento fuera de los límites

El desplazamiento está fuera de los límites. Consulte »Entrega de archivos basada en MQTT» sección en [AWS IoT Core Cuotas de servicio](#).

Se ha superado el límite de recuento de bloques

El número de bloques de solicitud está fuera de límites. Consulte »Entrega de archivos basada en MQTT» sección en [AWS IoT Core Cuotas de servicio](#).

Se ha superado el límite de mapa de bits de bloque

El tamaño del mapa de bits de solicitud está fuera de límites. Consulte »Entrega de archivos basada en MQTT» sección en [AWS IoT Core Cuotas de servicio](#).

ResourceNotFound

No se han encontrado la secuencia, los archivos, las versiones de archivos o los bloques solicitados. Consulte el mensaje de error para obtener más detalles.

VersionMismatch

La versión de transmisión de la solicitud no coincide con la versión de transmisión de la función de entrega de archivos basada en MQTT. Esto indica que los datos de transmisión se han modificado desde que el dispositivo recibió inicialmente la versión de transmisión.

EtagNo coincide

La ETag S3 de la transmisión no coincide con la ETag de la última versión de objeto S3.

InternalError

Se ha producido un error interno en la entrega de archivos basada en MQTT.

Ejemplo de caso de uso en FreeRTOS OTA

El agente FreeRTOS OTA (por aire) utiliza AWS IoT para transferir imágenes de firmware FreeRTOS a dispositivos FreeRTOS. Para enviar el conjunto de datos inicial a un dispositivo, utiliza el AWS IoT Servicio de Job para programar un trabajo de actualización de OTA en dispositivos FreeRTOS.

Para obtener una implementación de referencia de un cliente de entrega de archivos basado en MQTT, consulte [Códigos de agentes OTA FreeRTOS](#) en la documentación de FreeRTOS.

AWS IoT Device Defender

AWS IoT Device Defender es un servicio de seguridad que permite auditar la configuración de los dispositivos, monitorizar los dispositivos conectados para detectar un comportamiento anormal y mitigar los riesgos de seguridad. Le brinda la posibilidad de aplicar políticas de seguridad coherentes en toda su flota de dispositivos AWS IoT y responder rápidamente cuando los dispositivos sufren ataques.

Las flotas de IoT pueden constar de un gran número de dispositivos que tienen diversas funcionalidades, son de larga duración y están distribuidos geográficamente. Estas características hacen que la configuración de la flota sea compleja y propensa a errores. Y dado que los dispositivos a menudo están limitados en potencia informática, memoria y capacidades de almacenamiento, esto limita el uso del cifrado y otras formas de seguridad en los propios dispositivos. Además, los dispositivos a menudo usan software con vulnerabilidades conocidas. Estos factores hacen que las flotas de IoT sean un objetivo atractivo para los piratas informáticos y dificultan la protección continuada de la flota de dispositivos.

AWS IoT Device Defender fronta estos desafíos proporcionando herramientas para identificar los problemas de seguridad y las desviaciones de las prácticas recomendadas. AWS IoT Device Defender puede auditar las flotas de dispositivos para asegurarse de que cumplan con las prácticas recomendadas de seguridad y detecten un comportamiento anómalo en los dispositivos.

AWS Training and Certification

Realice el siguiente curso para empezar a usar AWS IoT Device Defender: [AWS IoT Device Defender Imprimación](#).

Introducción a AWS IoT Device Defender

Puede utilizar los siguientes aprendizajes para trabajar con AWS IoT Device Defender.

Temas

- [Configuración \(p. 871\)](#)
- [Guía de auditoría \(p. 873\)](#)
- [Guía de detección de ML \(p. 885\)](#)
- [Personaliza cuándo y cómo lo ves AWS IoT Device Defender resultados de auditoría \(p. 908\)](#)

Configuración

Antes de usar AWS IoT Device Defender por primera vez, realice las siguientes tareas:

- [Registrarse en AWS \(p. 871\)](#)
- [Creación de un usuario de IAM \(p. 872\)](#)

Estas tareas crean una Cuenta de AWS y un usuario de IAM con privilegios de administrador en la cuenta.

Registrarse en AWS

Al inscribirse en AWS, su cuenta se registra automáticamente en todos los servicios de AWS, como, por ejemplo, AWS IoT Device Defender. Si ya dispone de una Cuenta de AWS, pase a la siguiente tarea. Si no dispone de una Cuenta de AWS, utilice el siguiente procedimiento para crear una.

Si no dispone de una Cuenta de AWS, siga los pasos que figuran a continuación para crear una.

Para registrarse en Cuenta de AWS

1. Abra <https://portal.aws.amazon.com/billing/signup>.
2. Siga las instrucciones en línea.

Parte del procedimiento de inscripción consiste en recibir una llamada telefónica e indicar un código de verificación en el teclado del teléfono.

Anote su número de Cuenta de AWS ya que lo necesitará en la siguiente tarea.

Creación de un usuario de IAM

Este procedimiento describe cómo crearse usted mismo un usuario de IAM y agregarlo a un grupo con permisos administrativos de una política administrada asociada.

Para crearse usted mismo un usuario administrador y agregarlo a un grupo de administradores (consola)

1. Inicie sesión en la [consola de IAM](#) como el propietario de la cuenta; para ello, elija Root user (Usuario raíz) e ingrese el email de su Cuenta de AWS. En la siguiente página, escriba su contraseña.

Note

Le recomendamos que siga la práctica recomendada de utilizar el usuario de IAM **Administrator** como se indica a continuación y guardar de forma segura las credenciales del usuario raíz. Inicie sesión como usuario raíz únicamente para realizar algunas [tareas de administración de servicios y de cuentas](#).

2. En el panel de navegación, elija Usuarios y, a continuación, elija Agregar usuarios.
3. En User name (Nombre de usuario), escriba **Administrator**.
4. Seleccione la casilla de verificación situada junto a AWS Management Console access (Acceso a la consola). A continuación, seleccione Custom password (Contraseña personalizada) y luego escriba la nueva contraseña en el cuadro de texto.
5. (Opcional) De forma predeterminada, AWS requiere al nuevo usuario que cree una nueva contraseña la primera vez que inicia sesión. Puede quitar la marca de selección de la casilla de verificación situada junto a User must create a new password at next sign-in (El usuario debe crear una nueva contraseña en el siguiente inicio de sesión) para permitir al nuevo usuario restablecer su contraseña después de iniciar sesión.
6. Seleccione Next (Siguiente): Permisos.
7. En Set permissions (Establecer permisos), elija Add user to group (Añadir usuario a grupo).
8. Elija Create group (Crear grupo).
9. En el cuadro de diálogo Create group (Crear grupo), en Group name (Nombre del grupo) escriba **Administrators**.
10. Elija Filter policies (Filtrar políticas) y, a continuación, seleccione AWS managed - job function (Función de trabajo administrada por AWS) para filtrar el contenido de la tabla.
11. En la lista de políticas, active la casilla de verificación AdministratorAccess. A continuación, elija Create group (Crear grupo).

Note

Debe activar el acceso de usuarios y roles de IAM a Facturación para poder utilizar los permisos **AdministratorAccess** para acceder a la consola de AWS Billing and Cost

Management. Para ello, siga las instrucciones que se indican en el [paso 1 del tutorial sobre cómo delegar el acceso a la consola de facturación](#).

12. Retroceda a la lista de grupos y active la casilla de verificación del nuevo grupo. Elija Refresh si es necesario para ver el grupo en la lista.
13. Seleccione Next (Siguiente): Tags (Etiquetas).
14. (Opcional) Añadir metadatos al rol asociando las etiquetas como pares de clave-valor. Para obtener más información sobre el uso de etiquetas en IAM, consulte [Etiquetado de entidades de IAM](#) en la guía del usuario de IAM.
15. Seleccione Next (Siguiente): Review (Revisar)para ver la lista de suscripciones a grupos que se van a añadir al nuevo usuario. Cuando esté listo para continuar, elija Create user (Crear usuario).

Puede usar este mismo proceso para crear más grupos y usuarios, y para otorgar a los usuarios acceso a los recursos de la Cuenta de AWS. Para obtener información acerca de cómo usar las políticas que restringen los permisos de los usuarios a recursos de AWS específicos, consulte [Administración de accesos](#) y [Ejemplos de políticas](#).

Guía de auditoría

Este tutorial proporciona instrucciones sobre cómo configurar una auditoría periódica, configurar alarmas, revisar los resultados de la auditoría y mitigar los problemas de auditoría.

Temas

- [Requisitos previos \(p. 873\)](#)
- [Habilitación de comprobaciones \(p. 873\)](#)
- [Ver resultados de auditoría \(p. 877\)](#)
- [Creación de acciones de mitigación de auditoría \(p. 878\)](#)
- [Aplicar acciones de mitigación a los resultados de las auditorías \(p. 880\)](#)
- [Habilitar notificaciones SNS \(opcional\) \(p. 882\)](#)
- [Habilitar registro \(opcional\) \(p. 883\)](#)
- [Creación de un valorAWS IoT Device DefenderFunción de auditoría de IAM \(opcional\) \(p. 885\)](#)

Requisitos previos

Necesitará lo siguiente para completar este tutorial:

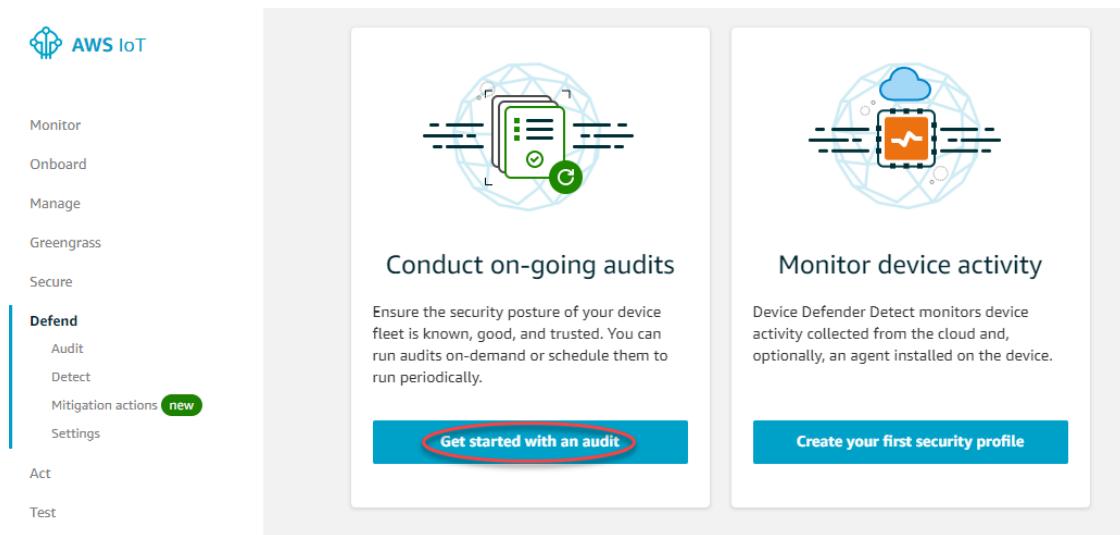
- Una Cuenta de AWS. Si no dispone de ella, consulte[Configuración de](#).

Habilitación de comprobaciones

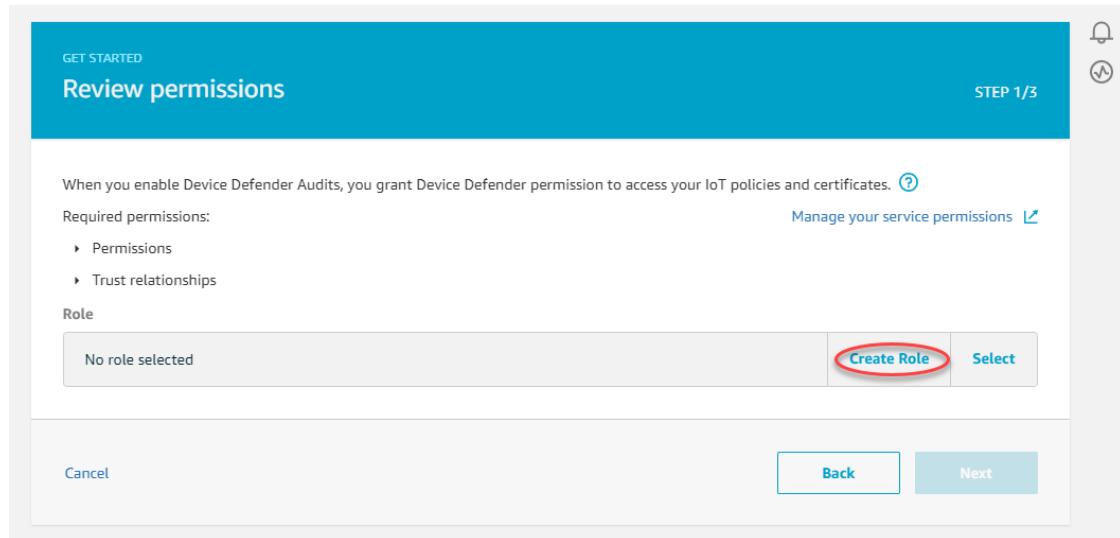
En el siguiente procedimiento, habilitará comprobaciones de auditoría que examinan la configuración y las políticas de cuenta y dispositivo para garantizar que existen medidas de seguridad. En este tutorial te indicamos que habilites todas las comprobaciones de auditoría, pero puedes seleccionar las comprobaciones que deseas.

El precio de auditoría es por recuento de dispositivos por mes (dispositivos de flota conectados aAWS IoT). Por lo tanto, agregar o quitar comprobaciones de auditoría no afectaría a su factura mensual al utilizar esta función.

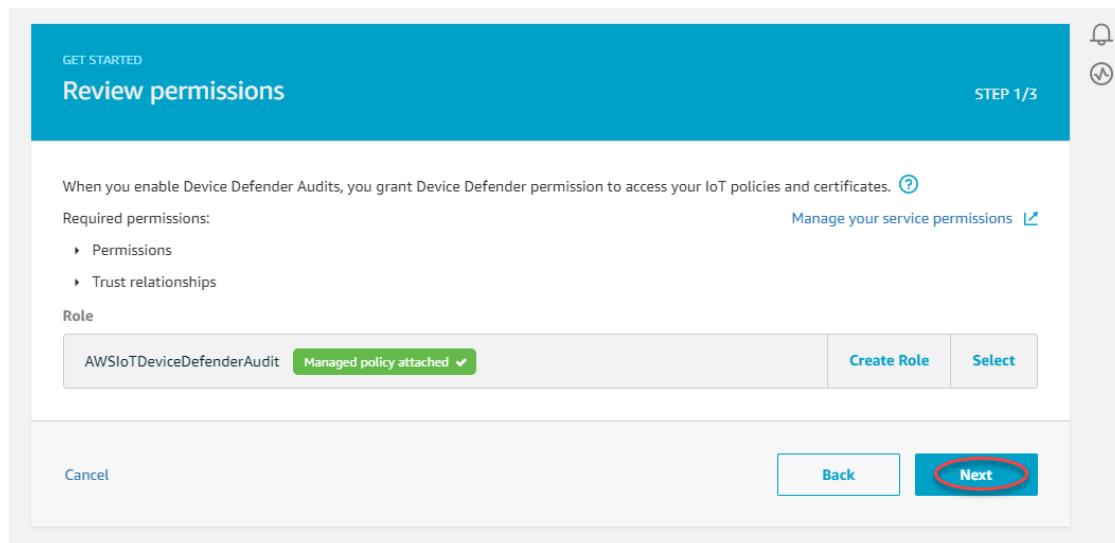
1. En el navegador[AWS IoTconsola](#), en el panel de navegación, expandaDefendery seleccioneIntroducción a una auditoría.



2. La introducción a Device Defender Auditofrece una descripción general de los pasos necesarios para habilitar las comprobaciones de auditoría. Una vez que haya revisado la pantalla, seleccione **Próximo**.
3. Si ya dispone de un rol, puede seleccionarlo. Si no, seleccione **Creación de un rol** y nombrelo **AWSIoTDeviceDefenderAudit**.



Debería ver los permisos necesarios asociados automáticamente al rol. Selecciona los triángulos junto a **Permisos** y **Relaciones de confianza** para ver qué permisos se conceden. Selecione **Próximo** cuando esté todo listo para seguir adelante.



4. En la página [Seleccionar cheques](#) verás todas las comprobaciones de auditoría que puedes seleccionar. Para este tutorial, le indicamos que seleccione todas las comprobaciones, pero puede seleccionar las comprobaciones que desee. Junto a cada comprobación de auditoría hay un ícono de ayuda que describe lo que hace la comprobación de auditoría. Para obtener más información acerca de auditorías, consulte [Comprobaciones de auditoría](#).

SelectPróximo una vez que hayas seleccionado tus cheques.

GET STARTED

Select checks

STEP 2/3

The checks you select here will be available when you set up audits. Data collection begins when a check has been enabled. All the free checks have been pre-selected for you. You can enable or disable checks at any time through the Device Defender Audit settings.

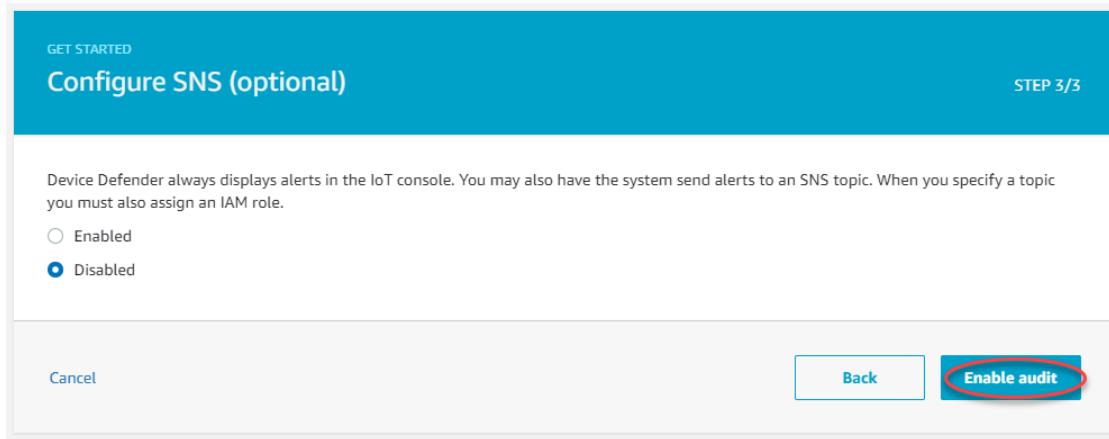
?

Check name	Severity	Resource type
<input checked="" type="checkbox"/> Authenticated Cognito role overly permissive ?	Critical	Cognito pool
<input checked="" type="checkbox"/> CA certificate key quality ?	Critical	CA certificate
<input checked="" type="checkbox"/> CA certificate revoked but device certificates still active ?	Critical	CA certificate
<input checked="" type="checkbox"/> Device Certificate key quality ?	Critical	Device certificate
<input checked="" type="checkbox"/> Device certificate shared ?	Critical	Device certificate
<input checked="" type="checkbox"/> IoT policies overly permissive ?	Critical	Policy
<input checked="" type="checkbox"/> Role Alias overly permissive ?	Critical	Role alias
<input checked="" type="checkbox"/> Unauthenticated Cognito role overly permissive ?	Critical	Cognito pool
<input checked="" type="checkbox"/> Conflicting MQTT client IDs ?	High	Client ID
<input checked="" type="checkbox"/> CA certificate expiring ?	Medium	CA certificate
<input checked="" type="checkbox"/> Device certificate expiring ?	Medium	Device certificate
<input checked="" type="checkbox"/> Revoked device certificate still active ?	Medium	Device certificate
<input checked="" type="checkbox"/> Role Alias allows access to unused services ?	Medium	Role alias
<input checked="" type="checkbox"/> Logging disabled ?	Low	Account settings

Cancel Back Next

Siempre puede cambiar sus comprobaciones de auditoría configuradas en Configuración.

5. En la página Configurar SNS (opcional) pantalla, selecciona Habilitar auditoría. Si desea habilitar las notificaciones SNS, consulte Habilitar notificaciones SNS (opcional) (p. 882).



6. Se te redirigirá a Schedules UNDER Auditoría.

Ver resultados de auditoría

En el siguiente procedimiento se muestra cómo ver los resultados de auditorías. En este tutorial, verá los resultados de auditoría de las comprobaciones de auditoría configuradas en [Habilitación de comprobaciones \(p. 873\)](#) "Hello, World!"

Para ver los resultados de auditoría

1. En el navegador [AWS IoT consola](#), en el panel de navegación, expanda **Defender**, seleccione **Auditoría** y seleccione **Resultados**.
2. La Resumen le informará si tiene cheques no conformes.

Name	Date	Status	Summary
On-demand	July 28, 2020, 14:14:18 (UTC-0700)	⚠️ Not compliant	1 of 14 non-compliant
On-demand	July 28, 2020, 11:55:43 (UTC-0700)	🟢 Compliant	14 of 14 completed
AWSIoTDeviceDefenderDailyAudit	July 28, 2020, 05:12:39 (UTC-0700)	⚠️ Not compliant	1 of 14 non-compliant
AWSIoTDeviceDefenderDailyAudit	July 27, 2020, 05:12:39 (UTC-0700)	⚠️ Not compliant	1 of 14 non-compliant
AWSIoTDeviceDefenderDailyAudit	July 26, 2020, 05:12:39 (UTC-0700)	⚠️ Not compliant	1 of 14 non-compliant
AWSIoTDeviceDefenderDailyAudit	July 25, 2020, 05:12:39 (UTC-0700)	⚠️ Not compliant	1 of 14 non-compliant
AWSIoTDeviceDefenderDailyAudit	July 24, 2020, 05:12:39 (UTC-0700)	⚠️ Not compliant	1 of 14 non-compliant
AWSIoTDeviceDefenderDailyAudit	July 23, 2020, 05:12:39 (UTC-0700)	⚠️ Not compliant	1 of 14 non-compliant
AWSIoTDeviceDefenderDailyAudit	July 22, 2020, 05:12:39 (UTC-0700)	⚠️ Not compliant	1 of 14 non-compliant
AWSIoTDeviceDefenderDailyAudit	July 21, 2020, 05:12:39 (UTC-0700)	⚠️ Not compliant	1 of 14 non-compliant

3. Seleccione el nombre de la comprobación de auditoría que quieras investigar.
`daily_audit_check - May 14, 2020 8:51:27 AM -0700`

Audit findings

Audit task ID 302ce82f9e33377e1f6be784532ff04c0 Started at May 14, 2020 8:51:27 AM -0700

▼ Non-compliant checks (2 of 14)

Check name	Severity	Non-compliant	% Resources	Mitigation
IoT policies overly permissive	Critical	1	50.0%	Use restrictive policies ⓘ
Logging disabled	Low	1	100%	Enable logging ⓘ

▼ Compliant checks (12 of 14)

▼ Mitigation actions

0 of 0

Created date	Task name	Status
You don't have any Audit action tasks yet.		

4. Utilice los signos de interrogación para obtener información sobre cómo hacer que sus comprobaciones no conformes. Por ejemplo, puede seguir [Habilitar registro \(opcional\) \(p. 883\)](#) para que la comprobación «Registro deshabilitado» sea compatible.

Creación de acciones de mitigación de auditoría

En el siguiente procedimiento, creará un AWS IoT Device Defender Acción de mitigación de auditorías para habilitar AWS IoT. Cada comprobación de auditoría tiene acciones de mitigación asignadas que afectarán a qué tipo de acción elige para la comprobación de auditoría que desea corregir. Para obtener más información, consulte [Acciones de mitigación](#).

Para utilizar la consola de AWS IoT para crear acciones de mitigación

1. Abra la [consola de AWS IoT](#).
2. En el panel de navegación de la izquierda, seleccione Defender y, a continuación, elija Mitigation Actions (Acciones de mitigación).
3. En la página Mitigation Actions (Acciones de mitigación), elija Create (Crear).

4. En la página Creación de una acción de mitigación En Action name (Nombre de acción), escriba un nombre único para su acción de mitigación, como **Habilitar acción de registro de errores**.
5. En Tipo de acción, elige Habilitación del registro de IoT.
6. En Rol de ejecución de acciones, seleccione Creación de un rol. Para Nombre, use **Función de registro de errores de acción de mitigación de IoT**. A continuación, elige Creación de un rol.
7. En Parámetros, en Rol de registro, seleccione AWSIoTLoggingRole. Para Log Level, elige Error.

Create a new mitigation action

You can use AWS IoT Device Defender to take actions to mitigate issues that were found during an audit. AWS IoT Device Defender provides predefined actions for the different audit checks. You can configure those actions for your AWS account and then apply them to a set of findings. [Learn more](#)

Action name [?](#)
EnableErrorLoggingAction

Action type [?](#)
Enable IoT logging

Permissions
Please create or select a role with the following mitigation action type specific permission(s) and trust relationship.
Required permissions:
▶ Permissions
▶ Trust relationships
[Manage your service permissions](#)

You can also attach an action specific managed policy to an existing role, or create a new role with the required managed policy attached.

Action execution role [?](#)
IoTMitigationActionErrorLoggingRole Managed policy attached ✓ [Create Role](#) [Select](#)

Parameters
Role for logging [?](#)
AWSIoTLoggingRole [Clear](#) [Select](#)

Log level [?](#)
Error

Tags
Tag name Provide a tag name, e.g. Manufacturer [Clear](#)
Value Provide a tag value, e.g. Acme-Corporation [Clear](#)
[Add another](#)

[Cancel](#) [Save](#)

8. Elija Guardar para guardar la acción de mitigación en su cuenta de AWS.
9. Una vez creada, verá la siguiente pantalla que indica que la acción de mitigación se ha creado correctamente.

The screenshot shows a confirmation message at the top: "Successfully created mitigation action". Below it is a table titled "Mitigation actions (1)". The table has columns: "Created date", "Action name", and "ARN". A single row is listed: "Jun 18, 2020 8:30:07 AM -0700", "EnableErrorLoggingAction", and "arn:aws:iot:us-east-1:765219403047:mitigationaction/EnableErrorLoggingAction". There are "Actions" and "Create" buttons at the top right of the table.

Mitigation actions (1)		
Created date	Action name	ARN
Jun 18, 2020 8:30:07 AM -0700	EnableErrorLoggingAction	arn:aws:iot:us-east-1:765219403047:mitigationaction/EnableErrorLoggingAction

Aplicar acciones de mitigación a los resultados de las auditorías

En el siguiente procedimiento se muestra cómo aplicar acciones de mitigación a los resultados de auditoría.

Para mitigar los hallazgos de auditoría no conformes

1. Abra la [consola de AWS IoT](#).
2. En el panel de navegación izquierdo, elija **Auditoría** y luego seleccione **Resultados**. Seleccione el nombre de la auditoría a la que desea responder.
3. Comprueba tus resultados. Observe que `Logging disabled` se encuentra en `Comprobaciones no conformes`.
4. Selecione **Iniciar acciones de mitigación**.

Device Defender > Audit > Results > AWSIoTDeviceDefenderDailyAudit

AWSIoTDeviceDefenderDailyAudit - Jun 16, 2020 11:23:17 PM -0700 Start mitigation actions

Audit findings

Audit task ID 08a38dcf887736bd947a95478646a3f1 Started at Jun 16, 2020 11:23:17 PM -0700

▼ Non-compliant checks (2 of 14)

Check name	Severity	Non-compliant	% Resources	Mitigation
IoT policies overly permissive	Critical	1	50.0%	Use restrictive p... ?
Logging disabled	Low	1	100%	Enable logging ?

► Compliant checks (12 of 14)

▼ Mitigation actions

0 of 0

Created date	Task name	Status
You don't have any Audit action tasks yet.		

5. UNDERSecciónar acciones, seleccione las acciones apropiadas para cada constatación no conforme a fin de resolver los problemas.

lucath@amazon.com - 049832161882 / Admin (Not Production Account) Pise

aws Services Oregon Support

START MITIGATION ACTION

Start a new mitigation action

Starting mitigation actions for the current audit report will start execution of the given actions against all findings in the report. If certain checks are not displayed in the list, it is because no actions have been configured with the type that is applicable for the check. [Create mitigation action](#)

Task name [?](#)

Select options for CA certificate revoked but device certificates still active ▲

Select actions [?](#)
1 actions selected Expand

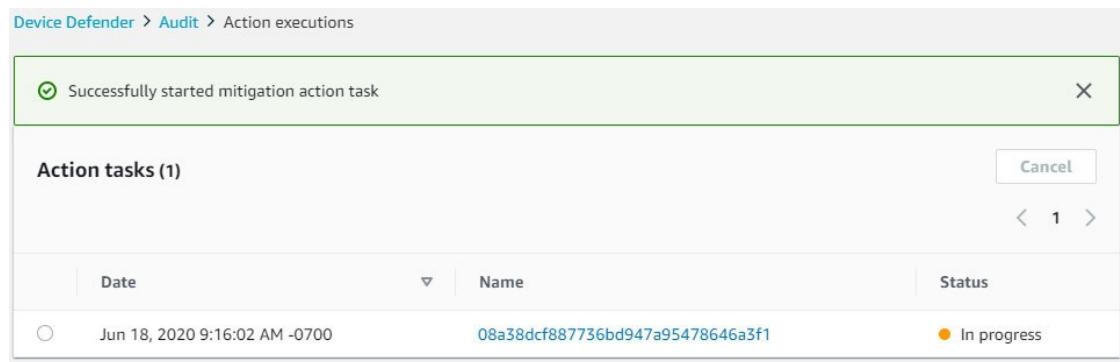
Select reason codes [?](#)
No reason codes selected Expand

Select options for IoT policies overly permissive ▲

Select actions [?](#)
1 actions selected Select all Close

AddressOverlyPermissivePolicies
 Empty_Policy

6. SelectConfirmar.
7. Una vez iniciada la acción de mitigación, puede tardar unos minutos en ejecutarse.



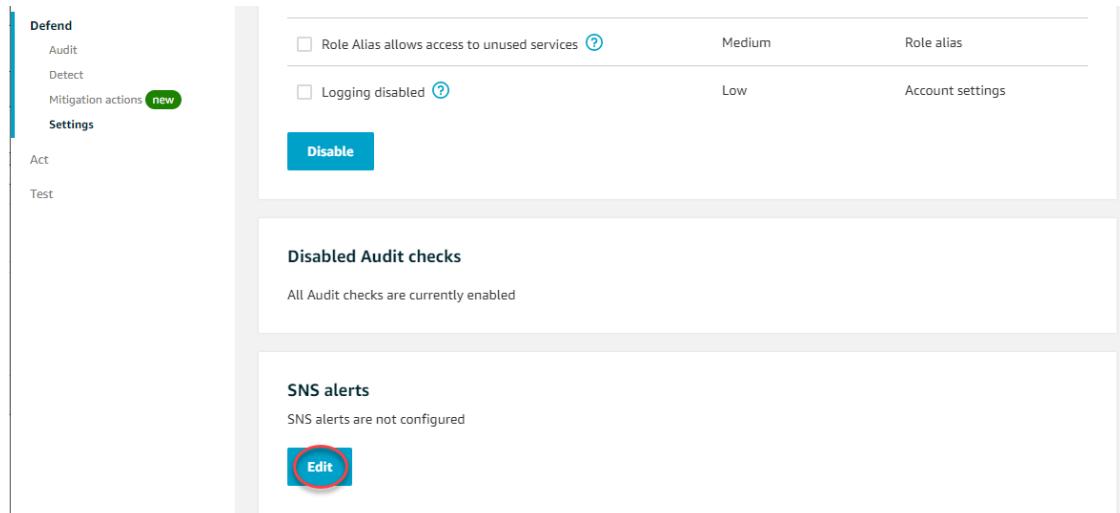
Para comprobar que la acción de mitigación ha funcionado

1. En el navegadorAWSconsola de, en el panel de navegación, seleccioneConfiguración.
2. Confirmar queRegistros sonEnabledy laGrado de detalle esError.

Habilitar notificaciones SNS (opcional)

En el siguiente procedimiento, habilita las notificaciones del Servicio de Notificaciones Simples (SNS) para avisarle cuando las auditorías identifican recursos no conformes. En este tutorial, configurará las notificaciones para las comprobaciones de auditoría habilitadas en elHabilitación de comprobaciones (p. 873)."Hello, World!"

1. En primer lugar, debe crear una política de IAM que proporcione acceso a Amazon SNS mediante elAWSConsola de administración. Para hacerlo, siga elCreación de un valorAWS IoT Device DefenderFunción de auditoría de IAM (opcional) (p. 885)proceso, pero seleccionandoAWSIoTDeviceDefenderPublishFindingsToSNSMitigationActionen el paso 8.
2. En el navegadorAWS IoTconsola, en el panel de navegación, expandaDefendery seleccioneConfiguración.
3. UNDERAlertas de SNS, seleccioneEditar.



4. En la páginaEdición de alertas de SNSpantalla, seleccionaEnabled (Habilitado). UNDERTema, seleccioneCrear. Asigne un nombre al temaNotificaciones IoT DDy seleccioneCrear. UNDERRol, seleccione el rol que ha creado llamadoAWSIoTDeviceDefenderAudit.

Edit SNS alerts

Device Defender always displays alerts in the IoT console. You may also have the system send alerts to an SNS topic. If you specify a topic you must also assign an IAM role.

Enabled
 Disabled

Topic

No topic selected	Create	Select
-------------------	------------------------	------------------------

Role

No role selected	Select
------------------	------------------------

[Cancel](#) [Update](#)

Seleccione Update (Actualizar).

Edit SNS alerts

Device Defender always displays alerts in the IoT console. You may also have the system send alerts to an SNS topic. If you specify a topic you must also assign an IAM role.

Enabled
 Disabled

Topic

IoTDDNotifications	Create	Clear	Select
--------------------	------------------------	-----------------------	------------------------

Role

AWSIoTDeviceDefenderAudit	Clear	Select
---------------------------	-----------------------	------------------------

[Cancel](#) [Update](#)

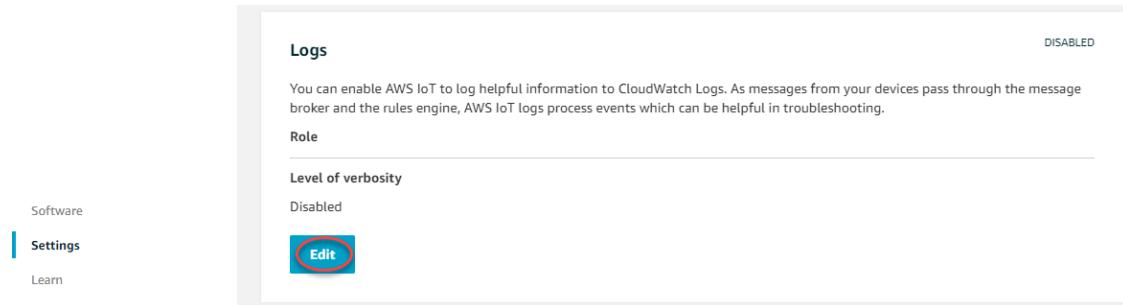
Si desea recibir correo electrónico o texto en sus plataformas de operaciones a través de SNS, consulte[Uso de Amazon SNS para notificaciones de usuario](#).

Habilitar registro (opcional)

Este procedimiento describe cómo habilitar AWS IoT para registrar información en CloudWatch Logs. Esto le permitirá ver los resultados de la auditoría. La habilitación del registro puede generar cargos incurridos.

Para habilitar el registro

1. En el navegador AWS consola de, en el panel de navegación, seleccione Configuración.
2. UNDER Registros, seleccione Editar.



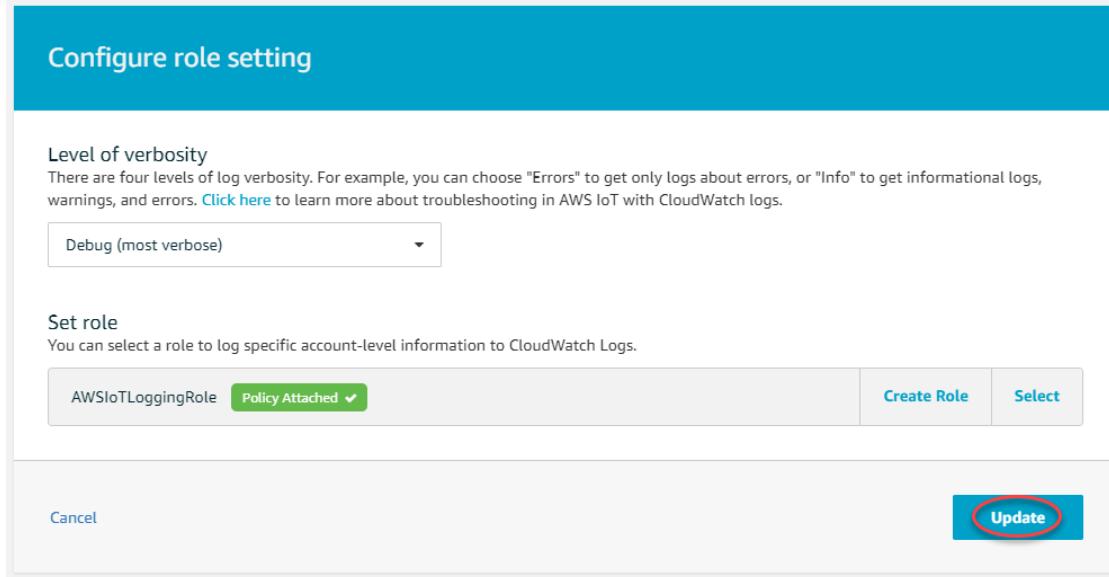
3. UNDER Grado de detalle, seleccione Depuración (más detallada).
4. UNDER Establecer rol, seleccione Creación de un rol y nombre al rol **Función de registro de AWS IOT**. Se adjuntará automáticamente una política.

The screenshot shows the 'Configure role setting' dialog box. It has a blue header bar with the title 'Configure role setting'. The main content area includes:

- Level of verbosity:** A dropdown menu set to 'Debug (most verbose)'.
- Set role:** A section where it says 'No role selected'. To its right are two buttons: 'Create Role' (which is circled in red) and 'Select'.

At the bottom of the dialog box are 'Cancel' and 'Update' buttons.

Seleccione Update (Actualizar).



Creación de un valor AWS IoT Device Defender Función de auditoría de IAM (opcional)

En el siguiente procedimiento, se crea un AWS IoT Device Defender Función de IAM de auditoría que proporciona acceso de lectura a AWS IoT.

1. Desplácese hasta la consola de IAM en <https://console.aws.amazon.com/iam/>
2. En el panel de navegación, elija Usuarios y luego elija Añada usuario.
3. En User name (Nombre de usuario), escriba **Administrator**.
4. Marque la casilla situada junto a Acceso a Management Console de. A continuación, seleccione Custom password (Contraseña personalizada) y luego escriba la nueva contraseña en el cuadro de texto.
5. (Opcional) De forma predeterminada, AWS requiere al nuevo usuario que cree una nueva contraseña la primera vez que inicia sesión. Puede quitar la marca de selección de la casilla de verificación situada junto a User must create a new password at next sign-in (El usuario debe crear una nueva contraseña en el siguiente inicio de sesión) para permitir al nuevo usuario restablecer su contraseña después de iniciar sesión.
6. Seleccione Next (Siguiente): Permisos.
7. En Set permissions (Establecer permisos), elija Attach existing policies directly (Asociar directamente las políticas existentes).
8. En la lista de políticas, active la casilla de verificación de AWSIoTDeviceDefenderAudit.
9. Seleccione Next (Siguiente): Tags (Etiquetas).
10. Seleccione Next (Siguiente): Review (Revisar) para ver la lista de suscripciones a grupos que se van a añadir al nuevo usuario. Cuando esté listo para continuar, elija Create user (Crear usuario).

Guía de detección de ML

En esta guía de introducción, creará un perfil de seguridad de detección de ML que utiliza aprendizaje automático (ML) para crear modelos de comportamiento esperado basados en datos de métricas históricos de sus dispositivos. Mientras ML Detect crea el modelo de ML, puede supervisar su progreso. Una vez

creado el modelo de ML, puede ver e investigar las alarmas de forma continua y mitigar los problemas identificados.

Para obtener más información sobre ML Detect y sus comandos API y CLI, consulte[Detección de ML \(p. 993\)](#).

El capítulo contiene las siguientes secciones:

- [Requisitos previos \(p. 886\)](#)
- [Cómo utilizar ML Detect en la consola \(p. 886\)](#)
- [Cómo utilizar ML Detect con la CLI \(p. 899\)](#)

Requisitos previos

- Una Cuenta de AWS. Si no dispone de esto, consulte[Configuración de](#).

Cómo utilizar ML Detect en la consola

Tutoriales

- [Habilitar detección de ML \(p. 886\)](#)
- [Supervisar el estado de su modelo ML \(p. 890\)](#)
- [Revise sus alarmas ML Detect \(p. 891\)](#)
- [Ajuste las alarmas ML \(p. 893\)](#)
- [Marcar el estado de verificación de la alarma \(p. 894\)](#)
- [Mitigar los problemas de dispositivos identificados \(p. 895\)](#)

Habilitar detección de ML

Los siguientes procedimientos detallan cómo configurar ML Detect en la consola de.

1. En primer lugar, asegúrate de que tus dispositivos crearán los puntos de datos mínimos necesarios tal como se definen en[Requisitos mínimos de ML \(p. 994\)](#)para la formación continua y la actualización del modelo. Para que la recopilación de datos progrese, asegúrese de que su perfil de seguridad esté asociado a un destino, que puede ser una cosa o un grupo de cosas.
2. En el navegador[AWS IoTconsola](#), en el panel de navegación, expandaDefender. ElegirDetectar,Perfiles de seguridad,Crear perfil de seguridad, y luegoCrear anomalía ML Detectar perfil.
3. En la páginaEstablecer configuraciones básicaspágina, haga lo siguiente:
 - UNDERObjetivo, elige los grupos de dispositivos de destino.
 - UNDERNombre del perfil de seguridad, escriba un nombre para su perfil de seguridad.
 - (Opcional) EnDescripciónPuede escribir una breve descripción del perfil ML.
 - UNDERComportamientos de métrica seleccionados en Security Profile, elige las métricas que te gustaría supervisar.

Metric	Type	ML Detect confidence	Datapoints required to trigger alarm	Datapoints required to clear alarm	Notifications
Authorization failures	Cloud-side	High	1	1	Suppressed
Connection attempts	Cloud-side	High	1	1	Suppressed
Disconnects	Cloud-side	High	1	1	Suppressed
Message size	Cloud-side	High	1	1	Suppressed
Messages received	Cloud-side	High	1	1	Suppressed
Messages sent	Cloud-side	High	1	1	Suppressed

Cuando haya terminado, elija Next.

- En la página **Establecer SNS (opcional)**, especifique un tema de SNS para las notificaciones de alarma cuando un dispositivo infringe un comportamiento de su perfil. Elija un rol de IAM que utilizará para publicar en el tema de SNS seleccionado.

Si todavía no tiene un rol de SNS, siga los siguientes pasos para crear un rol con los permisos y relaciones de confianza adecuados necesarios.

- Vaya a la [Consola de IAM](#). En el panel de navegación, seleccione Roles y, a continuación, seleccione Create role.
- UNDER Seleccione el tipo de entidad de confianza, seleccione AWS Service (Servicio). Luego, en Elija un caso de uso, eligelo Ty UNDER Seleccione su caso de uso, eligelo T - Acciones de mitigación de Device Defender. Cuando haya terminado, elija Siguiente: Permisos.
- UNDER Políticas de permisos adjuntos, asegurarse de que AWSIoTDeviceDefenderPublishFindingsToSNSSMitigationAction se selecciona y luego seleccione Siguiente: Tags (Etiquetas).

Create role

1 2 3 4

Attached permissions policies

The type of role that you selected requires the following policy.

Filter policies ▼			Search	Showing 6 results
Policy name ▼	Used as	Description		
▶ AWSIoTDeviceDefenderAddThingsToThingGrou...	Permissions policy (1)	Provides write access to IoT thing groups and r...		
▶ AWSIoTDeviceDefenderEnableIoTLoggingMitig...	Permissions policy (2)	Provides access for enabling IoT logging for ex...		
▶ AWSIoTDeviceDefenderPublishFindingsToSNS...	None	Provides messages publish access to SNS topi...		
▶ AWSIoTDeviceDefenderReplaceDefaultPolicyMi...	None	Provides write access to IoT policies for execut...		
▶ AWSIoTDeviceDefenderUpdateCACertMitigatio...	None	Provides write access to IoT CA certificates for ...		
▶ AWSIoTDeviceDefenderUpdateDeviceCertMitig...	None	Provides write access to IoT certificates for exe...		

Set permissions boundary

* Required

Cancel

Previous

Next: Tags

- UNDERAdd tags (opcional), puede añadir las etiquetas que desee asociar a su rol. Cuando haya terminado, elijaSiguiente: Consulte.
- UNDERReview (Revisar), dé nombre a su rol y asegúrese de queAWSIoTDeviceDefenderPublishFindingsToSNSSmitigationActionaparece en la listaPermisosyAWSservicio: iot.amazonaws.comaparece en la listaRelaciones de confianza. Cuando haya terminado, elijaCrear rol.

Identity and Access Management (IAM)

- Dashboard
- Access management
 - Groups
 - Users
 - Roles**
 - Policies
- Identity providers
- Account settings
- Access reports
 - Archive rules
 - Analyzers
 - Settings
- Credential report
- Organization activity
- Service control policies (SCPs)

Search IAM

Roles > Sample-SNS-role Summary Delete role

Role ARN	arn:aws:iam::049832161882:role/Sample-SNS-role				
Role description	Provides AWS IoT Device Defender write access to publish SNS notifications Edit				
Instance Profile ARNs					
Path	/				
Creation time	2020-12-21 17:13 PST				
Last activity	Not accessed in the tracking period				
Maximum session duration	1 hour Edit				
Permissions Trust relationships Tags Access Advisor Revoke sessions					
Permissions policies (1 policy applied)					
Attach policies + Add inline policy					
<table border="1"> <thead> <tr> <th>Policy name ▼</th> <th>Policy type ▼</th> </tr> </thead> <tbody> <tr> <td>▶ AWSIoTDeviceDefenderPublishFindingsToSNSSmitigationAction</td> <td>AWS managed policy</td> </tr> </tbody> </table>		Policy name ▼	Policy type ▼	▶ AWSIoTDeviceDefenderPublishFindingsToSNSSmitigationAction	AWS managed policy
Policy name ▼	Policy type ▼				
▶ AWSIoTDeviceDefenderPublishFindingsToSNSSmitigationAction	AWS managed policy				
Permissions boundary (not set)					

- En la página Editar comportamiento de métrica, puede personalizar la configuración de comportamiento de ML.

- Cuando haya terminado, elija Next.
- En la página Revisar configuración, verifique los comportamientos que desea supervisar el aprendizaje automático y, a continuación, elija Próximo.

Behavior name	Metric	Type	ML Detect confidence	Datapoints required to trigger alarm	Datapoints required to clear alarm	Notes
Authorization failures ML behavior	Authorization failures	Cloud-side	High	1	1	Sup
Bytes_out ML behavior	Bytes out	Device-side	High	1	1	Sup
Connection_attempts_M L_behavior	Connection attempts	Cloud-side	High	1	1	Sup
Disconnects _ML_behavior	Disconnects	Cloud-side	High	1	1	Sup

- Una vez que haya creado su perfil de seguridad, se lo redirigirá a Perfiles de seguridad de, donde aparece el perfil de seguridad recién creado.

Note

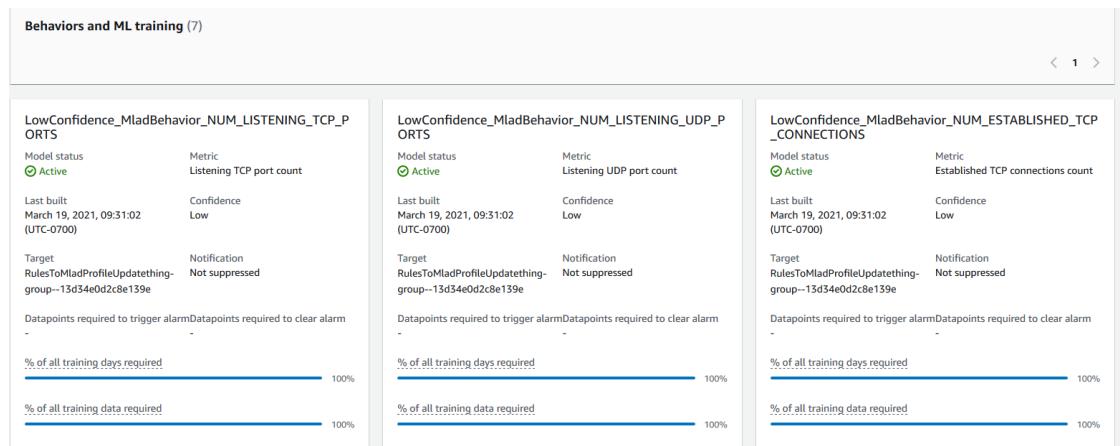
La formación y creación inicial del modelo ML tarda 14 días en completarse. Puede esperar ver las alarmas una vez finalizada, si hay alguna actividad anómala en sus dispositivos.

Supervisar el estado de su modelo ML

Mientras sus modelos de ML se encuentran en el período de formación inicial, puede supervisar su progreso en cualquier momento siguiendo los siguientes pasos.

- En el navegador [AWS IoT consola](#), en el panel de navegación, expanda **Defender** y luego seleccione **Detectar**, **Perfiles de seguridad**.
- En la página **Perfiles de seguridad de**, elige el perfil de seguridad que quieras revisar. A continuación, elige **Conductas y formación ML**.
- En la página **Conductas y formación ML**, comprueba el progreso de la formación de tus modelos de ML.

Después de que el estado de su modelo sea **Activo**, comenzará a tomar decisiones de **Detect** en función de tu uso y actualizará el perfil todos los días.



Note

Si tu modelo no progresá segúin lo esperado, asegúrate de que tus dispositivos cumplan con la[Requisitos mínimos \(p. 994\)](#).

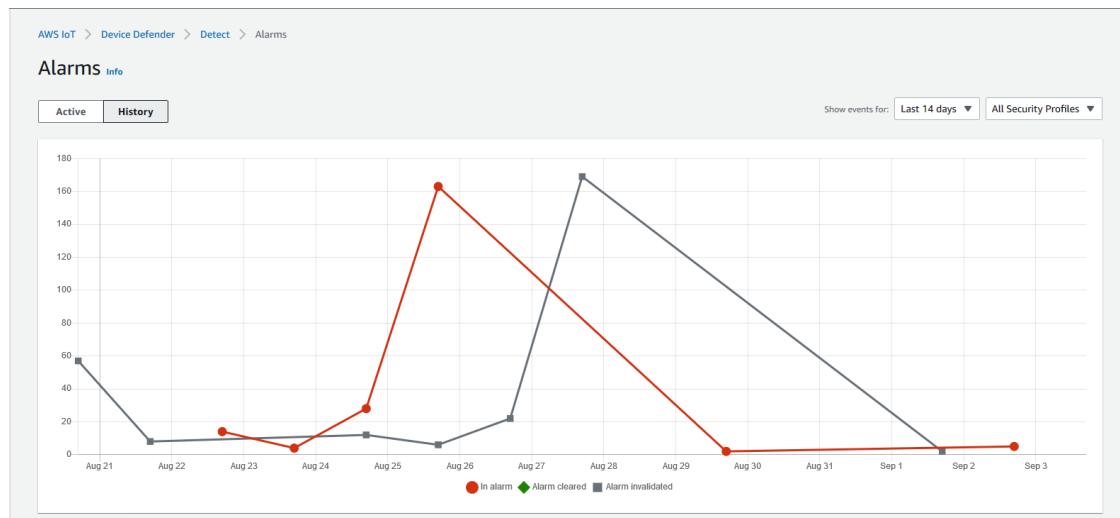
Revise sus alarmas ML Detect

Una vez que los modelos ML estén creados y listos para la inferencia de datos, puede ver e investigar periódicamente las alarmas identificadas por los modelos.

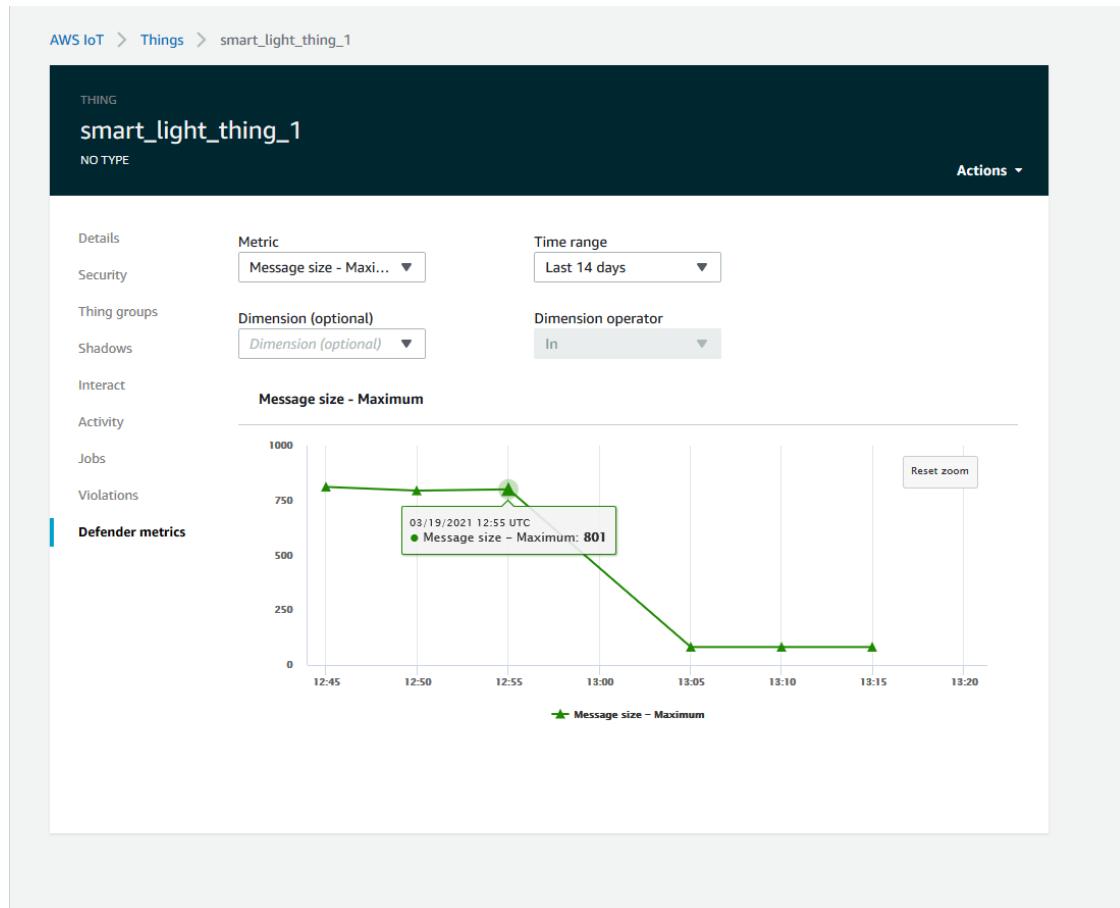
- En el navegador[AWS IoT consola](#), en el panel de navegación, expandaDefendery luego seleccioneDetectar,Alarmas.

First event	Thing name	Security Profile	Behavior type	Behavior name	Last emitted	Verification state	Confidence
September 03, 2021, 15:50:00 (UTC-0700)	iotconsole-6f8379bc-c245-4ffe-8ef7-b2b52e78975c	fdsa	Rule-based	Authorization_failures_behavior (Notification: on)	Authorization failures: 0 failure(s)	Unknown	-
September 03, 2021, 15:50:00 (UTC-0700)	iotconsole-539d0ef0-3504-4a9c-a7a1-be53b472b850	fdsa	Rule-based	Authorization_failures_behavior (Notification: on)	Authorization failures: 0 failure(s)	Unknown	-
September 03, 2021, 15:50:00 (UTC-0700)	iotconsole-81fc61d7-9362-4c87-ad6-333891ff7349	fdsa	Rule-based	Authorization_failures_behavior (Notification: on)	Authorization failures: 0 failure(s)	Unknown	-
September 03, 2021, 15:50:00 (UTC-0700)	iotconsole-23e8ec9a-2162-456e-a8c2-302c3826f618	fdsa	Rule-based	Authorization_failures_behavior (Notification: on)	Authorization failures: 0 failure(s)	Unknown	-
September 03, 2021, 15:50:00 (UTC-0700)	iotconsole-85e0278e-29aa-4554-b3b6-111b9063228f	fdsa	Rule-based	Authorization_failures_behavior (Notification: on)	Authorization failures: 0 failure(s)	Unknown	-

- Si navega hasta elHistoria, también puedes ver detalles sobre tus dispositivos que ya no están en alarmas.



Para obtener más información, en [ManejarelegirObjetos](#), elige la cosa para la que te gustaría ver más detalles y, a continuación, navega a [Métricas de Defender](#). Puede obtener acceso al [Gráfico de métricas de Defender](#) y lleve a cabo su investigación sobre cualquier cosa en alarma del [Activopestaña](#). En este caso, el gráfico muestra un aumento en el tamaño del mensaje, que activó la alarma. Posteriormente se puede ver la alarma desactivada.



Ajuste las alarmas ML

Una vez que los modelos de ML estén creados y listos para las evaluaciones de datos, puede actualizar la configuración de comportamiento de ML de su perfil de seguridad para cambiar la configuración. El siguiente procedimiento muestra cómo actualizar la configuración de comportamiento de ML de su perfil de seguridad en AWS CLI.

1. En el navegador [AWS IoT consola](#), en el panel de navegación, expanda **Device Defender** y luego seleccione **Detectar**, **Perfiles de seguridad**.
2. En la página **Perfiles de seguridad** de, seleccione la casilla situada junto al perfil de seguridad que desea revisar. A continuación, elige **Actions**, **Editar**.

Security Profile	Threshold type	Behaviors	Metrics retained	Target	Creation date	Notifications
Smart_lights_ML_Detect_Security_Profile	ML	9	-	All registered things	March 17, 2021, 12:58:14 (UTC-0700)	Suppressed (9)
MyEmptyGroupSP	ML	6	-	EmptyGroup	March 16, 2021, 17:52:01 (UTC-0700)	Suppressed (6)

3. UNDEREstablecer configuraciones básicas, puede ajustar los grupos de cosas de destino de Security Profile o cambiar las métricas que desea supervisar.

Set basic configurations Info

Select target and metrics that you would like to configure for your ML Security Profile.

Security Profile basic configuration

Target
Choose target device group(s)
All registered things

Security Profile name
Smart_lights_ML_Detect_Security_Profile
Enter a unique name containing only letters, numbers, hyphens, colon, or underscores. A Security Profile name cannot contain any spaces.

Description - optional
ML Detect security profile for monitoring smart lights

Selected metric behaviors in Security Profile (6) Info
You can assess how your fleet of devices is operating across the following metric behaviors.

Delete	Add cloud-side metric	Add device-side metric				
<input type="checkbox"/>	Metric	Type	ML Detect confidence	Datapoints required to trigger alarm	Datapoints required to clear alarm	Notifications
<input type="checkbox"/>	Authorization failures	Cloud-side	High	1	1	Suppressed
<input type="checkbox"/>	Connection attempts	Cloud-side	High	1	1	Suppressed
<input type="checkbox"/>	Disconnects	Cloud-side	High	1	1	Suppressed
<input type="checkbox"/>	Message size	Cloud-side	High	1	1	Suppressed
<input type="checkbox"/>	Messages received	Cloud-side	High	1	1	Suppressed
<input type="checkbox"/>	Messages sent	Cloud-side	High	1	1	Suppressed

4. Puede actualizar cualquiera de las opciones siguientes navegando a Editar comportamientos de métricas.
 - Los puntos de datos del modelo ML necesarios para activar la alarma
 - Los puntos de datos del modelo ML necesarios para borrar la alarma
 - Su nivel de confianza de ML Detect
 - Tus notificaciones de detección de ML (por ejemplo, No se ha suprimido, Suprimida)

The screenshot shows the 'Edit metric behaviors - optional' step in the AWS IoT Device Defender configuration process. It displays three sections: 'Authorization failures', 'Bytes out', and 'Connection attempts'. Each section has fields for Behavior name, Metric, Datapoints required to trigger alarm, Datapoints required to clear alarm, Notifications (set to 'Suppressed'), and ML Detect confidence (set to 'High').

Section	Behavior name	Metric	Datapoints required to trigger alarm	Datapoints required to clear alarm	Notifications	ML Detect confidence
Authorization failures	Authorization_failures_ML_behavior	Authorization failures	1	1	Suppressed	High
Bytes out	Bytes_out_ML_behavior	Bytes out	1	1	Suppressed	High
Connection attempts	Connection_attempts_ML_behavior	Connection attempts	1	1	Suppressed	High

Marcar el estado de verificación de la alarma

Marca tus alarmas estableciendo el estado de verificación y proporcionando una descripción de ese estado de verificación. Esto te ayuda a ti y a tu equipo a identificar alarmas a las que no tienes que responder.

1. En el navegador [AWS IoT consola](#), en el panel de navegación, expanda Defend y luego seleccione Detectar, Alarmas. Seleccione una alarma para marcar su estado de verificación.

The screenshot shows the AWS IoT Device Defender Detect Alarms interface. At the top, there are tabs for Active and History. Below is a table titled 'All alarms (1/5) Info' with columns: First event, Thing name, Security Profile, Behavior type, Behavior name, Last emitted, Verification state, and Confidence. One row is selected, showing details: First event (September 03, 2021, 15:50:00 UTC-0700), Thing name (iotconsole-6f8379bc-c245-4ffe-8ef7-b2b52e78975c), Security Profile (fdsa), Behavior type (Rule-based), Behavior name (Authorization_failures_behavior [Notification: on].....), Last emitted (September 03, 2021, 15:50:00 UTC-0700), Verification state (Unknown), and Confidence (-). There are buttons for 'Mark verification state' and 'Start mitigation actions' at the top right.

2. Elegir Marcar estado de verificación. Se abre el modal de estado de verificación.
3. Elija el estado de verificación adecuado, introduzca una descripción de verificación (opcional) y, a continuación, elija Mark. Esta acción asigna un estado de verificación y una descripción a la alarma elegida.

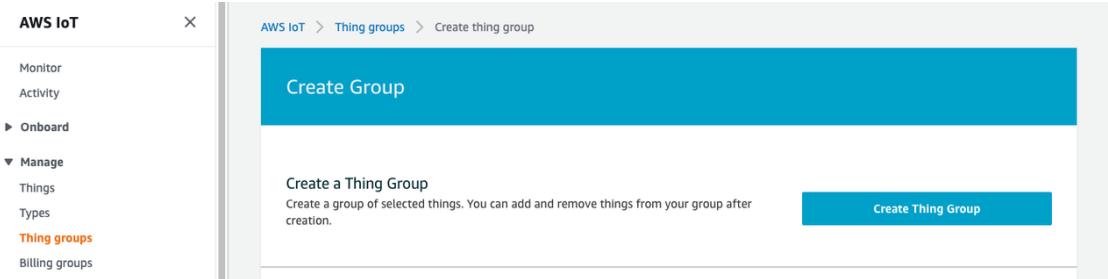
The screenshot shows the 'Mark verification state' modal dialog. It has a title bar with 'Mark verification state' and a close button. Below is a section titled 'Select verification state' with a note about helping AWS improve ML and Rules Detect features. A dropdown menu shows options: Unknown (selected), True positive, False positive, Benign positive, and Unknown. At the bottom are 'Cancel' and 'Mark' buttons, with 'Mark' being highlighted in orange.

Mitigar los problemas de dispositivos identificados

1. (Opcional) Antes de configurar acciones de mitigación de cuarentena, vamos a configurar un grupo de cuarentena al que trasladaremos el dispositivo que está infringiendo. También puede utilizar un grupo existente.
2. Vaya a Manejar Grupos de elementos, y luego Crear grupo de elementos. Nombra tu grupo de cosas. Para este tutorial, nombraremos nuestro grupo de cosas `quarantine_group`. UNDER Grupo de elementos, Seguridad, aplique la siguiente política al grupo de elementos.

```
{
  "Version": "2012-10-17",
```

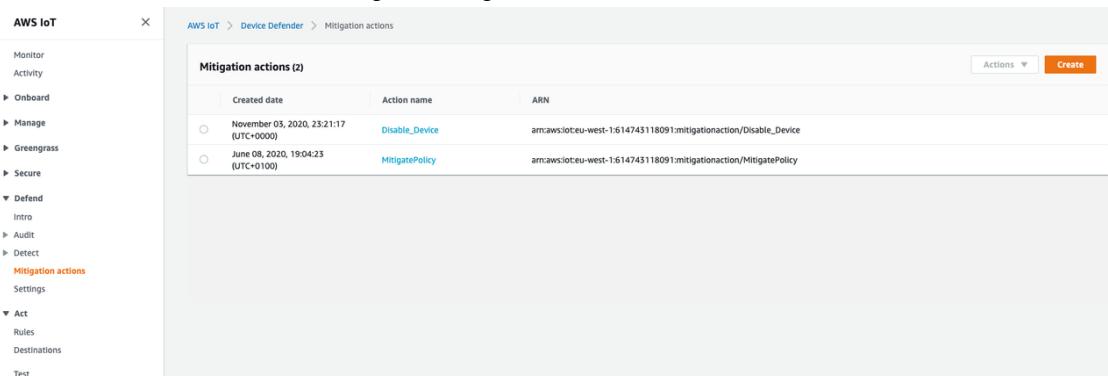
```
"Statement": [
    {
        "Effect": "Deny",
        "Action": "iot:/*",
        "Resource": "*"
    }
]
```



Cuando haya terminado, elija Crear grupo de elementos.

3. Ahora que hemos creado un grupo de cosas, vamos a crear una acción de mitigación que mueva dispositivos que, en alarma, al `Quarantine_group`.

UnderDefender, Acciones de mitigación, elige Crear.



4. En la página Crear una nueva acción de mitigación, escriba la siguiente información.
 - Action name (Nombre de acción): Dé un nombre a su acción de mitigación, como `Quarantine_action`.
 - Tipo de acción: Elija el tipo de acción. Vamos a elegir Agregar cosas al grupo de cosas (mitigación de auditoría o detección).
 - Rol de ejecución de acciones: Cree un rol o elija uno existente si lo creó anteriormente.
 - Parámetros: Elija un grupo de elementos. Podemos usar `Quarantine_group`, que hemos creado anteriormente.

Create a new mitigation action

You can use AWS IoT Device Defender to mitigate issues that were found during and audit or ongoing detect monitoring. There are predefined actions for the different audit checks and detect alarms to help you resolve issues quickly.

Action name [Info](#)

Action type [Info](#)

Permissions

Please create or select a role with the following mitigation action type specific permission(s) and trust relationship.

Required permissions: [Manage your service permissions ↗](#)

- ▶ Permissions
- ▶ Trust relationships

You can also attach an action specific managed policy to an existing role, or create a new role with the required managed policy attached.

Action execution role [Info](#)
 [Create Role](#) [Select](#)

Parameters

Thing groups [Info](#)

1 thing group(s) selected. [Close](#)

Thing groups	Summary
<input type="text" value="Quarantine_group"/>	

Cuando haya terminado, elija Save (Guardar). Ahora tiene una acción de mitigación que mueve dispositivos en alarma a un grupo de cosas de cuarentena y una acción de mitigación para aislar el dispositivo mientras investiga.

5. Vaya a **Defender**,**Detectar**,**Alarms**. Puede ver qué dispositivos están en estado de alarma en **Activo**.

First event	Thing name	Security Profile	Behavior type	Behavior name	Last emitted	Verification state	Confidence
September 03, 2021, 15:50:00 (UTC-0700)	iotconsole-6f8379bc-c245-4ffe-8ef7-b2b52e78975c	fdsa	Rule-based	Authorization_failures_behavior (Notification: on).....	Authorization failures: 0 failure(s)	Unknown	-
September 03, 2021, 15:50:00 (UTC-0700)	iotconsole-539d0ef0-3504-4a9c-a7a1-be53b472b850	fdsa	Rule-based	Authorization_failures_behavior (Notification: on).....	Authorization failures: 0 failure(s)	Unknown	-
September 03, 2021, 15:50:00 (UTC-0700)	iotconsole-01fc61d7-9362-4c87-ad6-333891ff7349	fdsa	Rule-based	Authorization_failures_behavior (Notification: on).....	Authorization failures: 0 failure(s)	Unknown	-
September 03, 2021, 15:50:00 (UTC-0700)	iotconsole-23e8ec9a-2162-456e-a8c2-302c3826f618	fdsa	Rule-based	Authorization_failures_behavior (Notification: on).....	Authorization failures: 0 failure(s)	Unknown	-
September 03, 2021, 15:50:00 (UTC-0700)	iotconsole-85e0278e-29aa-4554-b3b6-11b906323bf	fdsa	Rule-based	Authorization_failures_behavior (Notification: on).....	Authorization failures: 0 failure(s)	Unknown	-

Seleccione el dispositivo que desea mover al grupo de cuarentena y elija iniciar acciones de mitigación.

6. UNDER Iniciar acciones de mitigación, Iniciar acciones seleccione la acción de mitigación que creó anteriormente. Por ejemplo, vamos a elegir **Quarantine_action** y luego seleccione **Iniciar**. Se abre la página Tareas de acción.

Start mitigation actions

Select actions for mitigation.

Things effected by the selected alarm(s)
ddml7

Select Actions
The sequence of action executions follows the order of selected action(s)

Choose actions(s) to execute

Quarantine_action

I understand that the selected mitigation action(s) may not be reversible.

Cancel **Start**

7. El dispositivo está aislado en **Quarantine_group**, puede investigar la causa raíz del problema que activó la alarma. Después de completar la investigación, puede sacar el dispositivo del grupo de cosas o realizar más acciones.

Date	Task ID	Action name	Action type	Action parameter (1)	Action parameter (2)	Action Executions
December 02, 2020, 14:19:57 (UTCZ)	73fad2ea-9bd8-48d0-af3a-3dbc120b91e7	Quarantine_action	Add things to thing group	Thing group(s): Quarantine_group	Override dynamic groups: false	Successful

Cómo utilizar ML Detect con la CLI

A continuación se muestra cómo configurar ML Detect utilizando la CLI de.

Tutoriales

- [Habilitar detección de ML \(p. 899\)](#)
- [Supervisar el estado de su modelo ML \(p. 900\)](#)
- [Revise sus alarmas ML Detect \(p. 902\)](#)
- [Ajuste las alarmas ML \(p. 903\)](#)
- [Marcar el estado de verificación de la alarma \(p. 905\)](#)
- [Mitigar los problemas de dispositivos identificados \(p. 905\)](#)

Habilitar detección de ML

El siguiente procedimiento muestra cómo habilitar ML Detect enAWS CLI.

1. Asegúrese de que sus dispositivos crearán los puntos de datos mínimos necesarios tal como se define en[Requisitos mínimos de ML \(p. 994\)](#)para la formación continua y la actualización del modelo. Para que la recopilación de datos avance, asegúrate de que tus cosas se encuentren en un grupo de cosas adjunto a un perfil de seguridad.
2. Cree un perfil de seguridad de detección de ML mediante el[create-security-profile](#)comando. En el ejemplo siguiente se crea un perfil de seguridad denominado[*perfil de seguridad-para-luces inteligentes*](#)que comprueba el número de mensajes enviados, el número de errores de autorización, el número de intentos de conexión y el número de desconexiones. El ejemplo utilizamLDetectionConfigpara establecer que la métrica utilizará el 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"
}
}
}]'
```

```
"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
}
]'
```

Salida:

```
{
    "securityProfileName": "security-profile-for-smart-lights",
    "securityProfileArn": "arn:aws:iot:eu-west-1:123456789012:securityprofile/security-
profile-for-smart-lights"
}
```

3. A continuación, asocie su perfil de seguridad a uno o varios grupos de cosas. Usar `attach-security-profile` para asociar un grupo de elementos a su perfil de seguridad. En el ejemplo siguiente se asocia un grupo de cosas denominado `ML_DETECT_BETA_STATIC_GROUP` con `perfil de seguridad para luces inteligentes` Perfil de seguridad.

```
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
```

Salida:

Ninguno.

4. Después de crear el perfil de seguridad completo, el modelo ML comienza a entrenarse. La formación inicial del modelo ML y la construcción tardan 14 días en completarse. Después de 14 días, si hay actividad anómala en el dispositivo, puede esperar ver alarmas.

Supervisar el estado de su modelo ML

El siguiente procedimiento muestra cómo supervisar sus modelos de aprendizaje automático de formación en curso.

- Use `aws get-behavior-model-training-summaries` para ver el progreso del modelo de ML. En el siguiente ejemplo se obtiene el resumen del progreso de la formación del modelo de ML del *perfil de seguridad-para-luces inteligentes*. El `modelStatus` muestra si un modelo ha completado la formación o sigue pendiente de compilación para un comportamiento determinado.

```
aws iot get-behavior-model-training-summaries \
--security-profile-name security-profile-for-smart-lights
```

Salida:

```
{
  "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

Si tu modelo no progresá según lo esperado, asegúrate de que tus dispositivos cumplan con la[Requisitos mínimos \(p. 994\)](#).

Revise sus alarmas ML Detect

Una vez que los modelos ML estén creados y listos para las evaluaciones de datos, puede ver regularmente las alarmas que los modelos deduzcan. El siguiente procedimiento muestra cómo ver sus alarmas en elAWS CLI.

- Para ver todas las alarmas activas, utilice la[list-active-violations](#) comando.

```
aws iot list-active-violations \
--max-results 2
```

Salida:

```
{  
    "activeViolations": []  
}
```

Alternativamente, puede ver todas las infracciones detectadas durante un período de tiempo determinado mediante la[list-violation-events](#) comando. En el siguiente ejemplo se enumeran los eventos de infracción del 22 de septiembre de 2020 5:42:13 GMT al 26 de octubre de 2020 5:42:13 GMT.

```
aws iot list-violation-events \
--start-time 1599500533 \
--end-time 1600796533 \
--max-results 2
```

Salida:

```
{  
    "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": 1600780245.29  
        }  
    ]  
}
```

```
"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":
"Amo6XIUrsoohsojuIG6TuwSR3X9iUvH2OCksBZg6bed2j21VSnD1uP1pf1xKX1+a3cvBRSosIB0xFv40kM6RYBknZ/vxabMe/ZW31Ps/WiZHlr9Wg7R7eEGli59IJ/U0iBQ1McP/ht0E2XA2TTIvYeMmKQQPsRj/eov9j7P/wveu7skNGepU/mvpV0O2Ap7hnV5U+Prx/9+iJA/341va+pQww7jpUeHmJN9Hw4MqW0ysw0Ry3w38h0QWEpz2xwFWAxAARxeIxCxt5c37RK/lRZBlhYqoB+w2PZ74730h8pICGY4gktJxkwHyyRabpSM/G/f5DFrD905v8idkTZzBxW2jrbzSUIdafPtsZHL/yAMKr3HAKtaABz2nTsOBNrre7X2d/jIjjarhon0Dh91+8I9Y5Ey+DIFBcqFTvhibKAafQt3gs6CUiqHdWiencJyb8whmDE2qxvdxE1GmRb+k6kuN5jrZxxw95gzfYDgRHv11iEn8h1qZLD0czk1FBpMppHj9cetHPvM+qffXGAzKi8tL6eQuCdMLxmVE3jbqcjcjk9ItnaYJi5zKDz9FVbrz9qZZPtZJFHp"
}
```

Ajuste las alarmas ML

Una vez que los modelos de ML estén creados y listos para las evaluaciones de datos, puede actualizar la configuración de comportamiento de ML de su perfil de seguridad para cambiar la configuración. El siguiente procedimiento muestra cómo actualizar la configuración de comportamiento de ML de su perfil de seguridad en AWS CLI.

- Para cambiar la configuración de comportamiento de ML de su perfil de seguridad, utilice la `update-security-profile` comando. En el siguiente ejemplo se actualiza el **perfil de seguridad-para-luces inteligentes** Comportamientos del perfil de seguridad cambiando el `confidenceLevel` de algunos de los comportamientos y notificaciones sin supresión de todos los comportamientos.

```
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
}
]'
```

Salida:

```
{
  "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
}
```

Marcar el estado de verificación de la alarma

Puedes marcar tus alarmas con estados de verificación para ayudar a clasificar las alarmas e investigar anomalías.

- Marca tus alarmas con un estado de verificación y una descripción de ese estado. Por ejemplo, para establecer el estado de verificación de una alarma en Falso positivo, utilice el siguiente 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
```

Salida:

Ninguno.

Mitigar los problemas de dispositivos identificados

1. Usar `create-thing-group` para crear un grupo de elementos para la acción de mitigación. En el siguiente ejemplo, creamos un grupo de cosas llamado Grupo de cosas para la acción Detect Mitigation.

```
aws iot create-thing-group --thing-group-name ThingGroupForDetectMitigationAction
```

Salida:

```
{
    "thingGroupName": "ThingGroupForDetectMitigationAction",
    "thingGroupArn": "arn:aws:iot:us-east-1:123456789012:thinggroup/ThingGroupForDetectMitigationAction",
    "thingGroupId": "4139cd61-10fa-4c40-b867-0fc6209dca4d"
}
```

2. A continuación, utilice el `create-mitigation-action` para crear una acción de mitigación. En el siguiente ejemplo, creamos una acción de mitigación llamada detect_mitigation_action con el ARN del rol de IAM que se utiliza para aplicar la acción de mitigación. También definimos el tipo de acción y los parámetros de dicha acción. En este caso, nuestra mitigación trasladará las cosas a nuestro grupo de cosas creado anteriormente llamado Grupo de cosas para la acción Detect Mitigation.

```
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
    }
}'
```

```
}
```

Salida:

```
{  
    "actionArn": "arn:aws:iot:us-  
east-1:123456789012:mitigationaction/detect_mitigation_action",  
    "actionId": "5939e3a0-bf4c-44bb-a547-1ab59ffe67c3"  
}
```

3. Usar `start-detect-mitigation-actions-task` para iniciar la tarea de acciones de mitigación. `task-id`, `target` y `actions` son parámetros obligatorios.

```
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
```

Salida:

```
{  
    "taskId": "taskIdForMitigationAction"  
}
```

4. (Opcional) Para ver las ejecuciones de acciones de mitigación incluidas en una tarea, utilice la `list-detect-mitigation-actions-executions` comando.

```
aws iot list-detect-mitigation-actions-executions \  
    --task-id taskIdForMitigationAction \  
    --max-items 5 \  
    --page-size 4
```

Salida:

```
{  
    "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) Utilice el `describe-detect-mitigation-actions-task` para obtener información sobre una tarea de acción de mitigación.

```
aws iot describe-detect-mitigation-actions-task \  
    --task-id taskIdForMitigationAction
```

Salida:

```
{  
    "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/MitigationActionValidRole",  
                "actionParams": {  
                    "addThingsToThingGroupParams": {  
                        "thingGroupNames": [  
                            "ThingGroupForDetectMitigationAction"  
                        ],  
                        "overrideDynamicGroups": false  
                    }  
                }  
            ]  
        ],  
        "taskStatistics": {  
            "actionsExecuted": 0,  
            "actionsSkipped": 0,  
            "actionsFailed": 0  
        }  
    }  
}
```

6. (Opcional) Para obtener una lista de las tareas de acciones de mitigación, utilice el `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
```

Salida:

```
{  
    "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/MitigationActionValidRole",  
                    "actionParams": {  
                        "addThingsToThingGroupParams": {  
                            "thingGroupNames": [  
                                "ThingGroupForDetectMitigationAction"  
                            ],  
                            "overrideDynamicGroups": false  
                        }  
                    }  
                ]  
            ],  
            "taskStatistics": {  
                "actionsExecuted": 0,  
                "actionsSkipped": 0,  
                "actionsFailed": 0  
            }  
        }  
    ]  
}
```

```
        "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 una tarea de acciones de mitigación, utilice la[cancel-detect-mitigation-actions-task](#) comando.

```
aws iot cancel-detect-mitigation-actions-task \
--task-id taskIdForMitigationAction
```

Salida:

Ninguno.

Personaliza cuándo y cómo lo vesAWS IoT Device Defenderresultados de auditoría

AWS IoT Device Defenderauditoría proporciona comprobaciones de seguridad periódicas para confirmarAWS IoT dispositivos y recursos siguen las prácticas recomendadas. Para cada comprobación, los resultados de la auditoría se clasifican como conformes o no conformes, donde el incumplimiento da como resultado iconos de advertencia de consola. Para reducir el ruido causado por problemas conocidos repetidos, la función de supresión de búsqueda de auditoría le permite silenciar temporalmente estas notificaciones de incumplimiento.

Puede suprimir las comprobaciones de auditoría seleccionadas de un recurso o cuenta específicos durante un período de tiempo predeterminado. Un resultado de comprobación de auditoría que se ha suprimido se clasifica como una constatación suprimida, independiente de las categorías conformes y no conformes. Esta nueva categoría no activa una alarma como un resultado incompatible. Esto le permite reducir las perturbaciones de las notificaciones de incumplimiento durante períodos de mantenimiento conocidos o hasta que esté programada la finalización de una actualización.

Introducción

En las siguientes secciones se detalla cómo utilizar las supresiones de búsqueda de auditoría para suprimir un `Device certificate expiring` en la consola y la CLI. Si quieras seguir alguna de las demostraciones, primero debes crear dos certificados caducados para que Device Defender los detecte.

Utilice lo siguiente para crear los certificados.

- [Crear y registrar un certificado de entidad de certificación \(p. 302\)](#)
- [Creación de un certificado de cliente mediante el certificado de entidad de certificación](#). En el paso 3, establezca `sudays` parámetro para 1.

Si utiliza la CLI para crear certificados, introduzca el siguiente comando.

```
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
```

Personalizar los hallazgos de auditoría en la consola

En el siguiente tutorial se utiliza una cuenta con dos certificados de dispositivo caducados que desencadenan una comprobación de auditoría no conforme. En este escenario, queremos deshabilitar la advertencia porque nuestros desarrolladores están probando una nueva función que solucionará el problema. Creamos una supresión de búsqueda de auditoría para cada certificado para evitar que el resultado de la auditoría no sea conforme durante la próxima semana.

1. En primer lugar, realizaremos una auditoría bajo demanda para demostrar que la comprobación del certificado de dispositivo caducado no cumple las normas.

Desde la [AWS IoT consola](#), elige `Device Defender` desde la barra lateral izquierda, luego `Auditoría`, y luego `Resultados`. En la página `Resultados de auditoría`, elija `Crear`. La `Crear una nueva auditoría` se abrirá una ventana. Elija `Create (Crear)`.

▼ Defend

Intro

▼ Audit

Results

Schedules

Action executions

Finding suppressions

new

► Detect

A partir de los resultados de la auditoría bajo demanda, podemos ver que el «certificado de dispositivo caduca» no cumple los requisitos para dos recursos.

2. Ahora, nos gustaría deshabilitar la advertencia de comprobación de no conformidad «Certificado de dispositivo que caduca» porque nuestros desarrolladores están probando nuevas características que solucionarán la advertencia.

Desde la barra lateral izquierda debajoDefender, eligeAuditoríay luego seleccioneBuscar supresiones. En la páginaSupresiones de hallazgos de auditoría, elijaCrear.

POLICIES

CAs

Role Aliases

Authorizers

▼ Defend

Intro

▼ Audit

Results

Schedules

Action executions

Finding suppressions

3. En la página Crear una supresión de búsqueda de auditoría ventana, tenemos que rellenar lo siguiente.
 - Comprobación de auditoría: Seleccionamos Device certificate expiring, porque esa es la comprobación de auditoría que nos gustaría suprimir.
 - Identificador de recursos: Introdujimos el identificador del certificado del dispositivo de uno de los certificados para los que nos gustaría suprimir los resultados de auditoría.
 - Duración de supresión: Seleccionamos 1 week, porque durante cuánto tiempo nos gustaría suprimir el Device certificate expiring comprobación de auditoría para.
 - Descripción (opcional): Añadimos una nota que describe por qué estamos suprimiendo este hallazgo de auditoría.

Create an audit finding suppression

Suppressing an audit finding on a specified resource means that AWS IoT Device Defender will no longer consider the device non-compliant.

Audit check

Device certificate expiring

Resource identifier

Device certificate id

b4490bd64c5cf85182f3182f1c03e70017e483f17b

Suppression duration

1 week

Description (optional)

Developer updates

Una vez llenados los campos, elijaCrear. Vemos un banner de éxito después de que se haya creado la supresión del hallazgo de auditoría.

4. Hemos suprimido un hallazgo de auditoría de uno de los certificados y ahora tenemos que suprimir el hallazgo de auditoría del segundo certificado. Podríamos utilizar el mismo método de supresión que usamos en el paso 3, pero utilizaremos otro método para fines de demostración.

Desde la barra lateral izquierda debajoDefender, eligeAuditoríay luego seleccioneResultados. En la páginaResultados de auditoría, elija la auditoría con el recurso no conforme. A continuación, seleccione el recurso enComprobaciones no conforme. En nuestro caso, seleccionamos «El certificado del dispositivo caduca».

5. En la páginaEl certificado de dispositivo caducadopágina, enPolítica no conformeelija el botón de opción situado junto a la hallazgo que debe suprimirse. A continuación, elija laActionsmenú desplegable y, a continuación, elija la duración durante la que desea que se suprima. En nuestro caso, elegimos1 weekcomo lo hicimos para el otro certificado. En la páginaConfirmar supresión, elijaActivar supresión.

2 OUT OF 195 device certificates non-compliant

Mitigation

Consult your security best practices for how to proceed.

1. Provision a new certificate and attach it to the device.
2. Verify that the new certificate is valid and the device can connect.
3. Mark the old certificate as "INACTIVE" in the AWS IoT console.
4. Detach the old certificate from the device. (See [Detaching a certificate](#))

Non-compliant certificate (2)

Finding



28022a890964e991852c79a28a83eb89



dc9b109c705ed7e68588bc54eef86f1c

Vemos un banner de éxito después de que se haya creado la supresión del hallazgo de auditoría. Ahora, ambos resultados de auditoría se han suprimido durante 1 semana mientras nuestros desarrolladores trabajan en una solución para abordar la advertencia.

Personalizar los resultados de auditoría en la CLI

En el siguiente tutorial se utiliza una cuenta con un certificado de dispositivo caducado que desencadena una comprobación de auditoría no conforme. En este escenario, queremos deshabilitar la advertencia porque nuestros desarrolladores están probando una nueva función que solucionará el problema. Creamos una supresión de búsqueda de auditoría para el certificado para evitar que el resultado de la auditoría no sea conforme durante la próxima semana.

Utilizamos los siguientes comandos de CLI.

- [create-audit-suppression](#)
- [describe-audit-supression](#)
- [update-audit-supression](#)
- [supresión de eliminación de auditorías](#)
- [lista-auditorías-supresiones](#)

1. Utilice el siguiente comando para habilitar la auditoría.

```
aws iot update-account-audit-configuration \
    --audit-check-configurations "{\"DEVICE_CERTIFICATE_EXPIRING_CHECK\":{\"enabled \
    \":true}}"
```

Salida:

Ninguno.

2. Utilice el siguiente comando para ejecutar una auditoría bajo demanda dirigida a la DEVICE_CERTIFICATE_EXPIRING_CHECK Comprobación de auditoría.

```
aws iot start-on-demand-audit-task \
    --target-check-names DEVICE_CERTIFICATE_EXPIRING_CHECK
```

Salida:

```
{ \
    "taskId": "787ed873b69cb4d6cdbae6ddd06996c5" \
}
```

3. Usar [describe-account-audit-configuration](#) para describir la configuración de auditoría. Queremos confirmar que hemos activado la comprobación de auditoría de DEVICE_CERTIFICATE_EXPIRING_CHECK.

```
aws iot describe-account-audit-configuration
```

Salida:

```
{ \
    "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": {  
    "AUTENTICATED_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 debería tener un valor de true.

4. Usar [lista de tareas de auditoría](#) para identificar las tareas de auditoría completadas.

```
aws iot list-audit-tasks \  
    --task-status "COMPLETED" \  
    --start-time 2020-07-31 \  
    --end-time 2020-08-01
```

Salida:

```
{  
    "tasks": [  
        {  
            "taskId": "787ed873b69cb4d6cdbae6ddd06996c5",  
            "taskStatus": "COMPLETED",  
            "taskType": "SCHEDULED_AUDIT_TASK"  
        }  
    ]  
}
```

La `taskId` de la auditoría que ejecutó en el paso 1 debe tener un `taskStatus` de COMPLETED.

5. Usar [describe-audit-task](#) para obtener detalles acerca de la auditoría completada mediante la `taskId` salida del paso anterior. Este comando enumera los detalles de la auditoría.

```
aws iot describe-audit-task \  
--task-id "787ed873b69cb4d6cdbae6ddd06996c5"
```

Salida:

```
{  
    "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 [list-auditorías-hallazgos](#) para buscar el identificador del certificado no conforme para que podamos suspender las alertas de auditoría de este recurso.

```
aws iot list-audit-findings \  
--start-time 2020-07-31 \  
--end-time 2020-08-01
```

Salida:

```
{  
    "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 [create-audit-supression](#) para suprimir las notificaciones del DEVICE_CERTIFICATE_EXPIRING_CHECK comprobación de auditoría de un certificado de dispositivo con el id [c7691e<shortened>](#) hasta [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 [supresión de auditorías de lista](#) para confirmar la configuración de supresión de auditoría y obtener detalles sobre la supresión.

```
aws iot list-audit-suppressions
```

Salida:

```
{  
    "suppressions": [  
        {  
            "checkName": "DEVICE_CERTIFICATE_EXPIRING_CHECK",  
            "resourceIdentifier": {  
                "deviceCertificateId": "c7691e<shortened>"  
            },  
            "expirationDate": 1597881600.0,
```

```
        "suppressIndefinitely": false
    }
}
```

9. La [update-audit-supression](#) se puede utilizar para actualizar la supresión de la búsqueda de auditoría. En el siguiente ejemplo se actualizan las `expiration-date` a 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. La [supresión de eliminación de auditorías](#) se puede utilizar para eliminar una supresión de búsqueda de auditoría.

```
aws iot delete-audit-suppression \
--check-name DEVICE_CERTIFICATE_EXPIRING_CHECK \
--resource-identifier deviceCertificateId="c7691e<shortened>"
```

Para confirmar la eliminación, utilice el comando [list-audit-suppressions](#).

```
aws iot list-audit-suppressions
```

Salida:

```
{
  "suppressions": []
}
```

En este tutorial, te mostramos cómo suprimir un `Device certificate expiring` comprobando en la consola y la CLI. Para obtener más información acerca de las supresiones de hallazgos de auditoría, consulte [Supresiones de hallazgos de auditoría \(p. 974\)](#)

Auditoría

Una auditoría de AWS IoT Device Defender examina la configuración y las políticas relacionadas con una cuenta y dispositivo para garantizar que existen medidas de seguridad. Las auditorías pueden ayudarle a detectar cualquier desviación con respecto a las prácticas recomendadas de seguridad o a las políticas de acceso adecuadas (por ejemplo, cuando varios dispositivos usan la misma identidad o cuando existen políticas excesivamente permisivas que permiten a un dispositivo leer y actualizar datos de muchos otros dispositivos). Puede ejecutar auditorías según sean necesarias (auditorías bajo demanda) o programarlas para que se ejecuten periódicamente (auditorías programadas).

Una auditoría de AWS IoT Device Defender realiza una serie de comprobaciones predefinidas relacionadas con las prácticas recomendadas de seguridad de IoT comunes y las vulnerabilidades de los dispositivos. Entre las comprobaciones se incluyen las políticas que conceden permiso para leer o actualizar datos en varios dispositivos, los dispositivos que comparten una identidad (certificado X.509), o los certificados que van a caducar o que se han revocado pero siguen activos.

Gravedad del problema

La gravedad del problema indica el nivel de gravedad asociado con cada caso identificado de incumplimiento y el tiempo recomendado para remediarlo.

Critical

Las comprobaciones de auditoría no conformes con esta gravedad identifican los problemas que requieren atención inmediata. Los problemas críticos permiten a menudo a los agentes malintencionados obtener fácilmente acceso a sus recursos o controlarlos sin demasiada sofisticación y sin disponer de información privilegiada o credenciales especiales.

Alta

Las comprobaciones de auditoría no conformes con esta gravedad requieren una investigación urgente y una planificación de remediación después de que se resuelvan los problemas críticos. Al igual que los problemas críticos, los problemas con gravedad alta suelen proporcionar los usuarios malintencionados acceso o control de sus recursos. Sin embargo, los problemas con gravedad alta suelen ser más difíciles de aprovechar. Es posible que requieran herramientas especiales, información privilegiada o configuraciones específicas.

Media

Las comprobaciones de auditoría no conformes con esta gravedad presentan problemas que requieren atención como parte del mantenimiento continuo del estado de seguridad. Los problemas de gravedad media pueden causar un impacto operativo negativo, como interrupciones no planificadas debido a un mal funcionamiento de los controles de seguridad. Estos problemas también pueden proporcionar a los usuarios malintencionados acceso o control limitado a sus recursos, o pueden facilitar algunos pasos de sus acciones maliciosas.

Baja

Las comprobaciones de auditoría no conformes con esta gravedad suelen indicar que las prácticas recomendadas de seguridad se descuidaron o se pasaron por alto. Aunque pueden no causar un impacto inmediato en la seguridad, estos descuidos pueden ser aprovechados por usuarios con malas intenciones. Al igual que los problemas de gravedad media, los problemas de gravedad baja requieren atención como parte del mantenimiento continuo del estado de seguridad.

Pasos siguientes

Para conocer los tipos de comprobaciones de auditoría que se pueden realizar, consulte [Comprobaciones de auditoría \(p. 920\)](#). Para obtener información sobre las cuotas de servicio que se aplican a las auditorías, consulte la sección [Cuotas de servicio](#).

Comprobaciones de auditoría

Note

Cuando habilita una comprobación, la recopilación de datos comienza inmediatamente. Si tiene que recopilar una gran cantidad de datos en la cuenta, es posible que los resultados de la comprobación no estén disponibles durante cierto tiempo después de habilitarlos.

Es posible realizar las siguientes comprobaciones de auditoría:

- [Certificado de CA revocado pero los certificados de dispositivo siguen activos \(p. 921\)](#)
- [Certificado de dispositivo compartido \(p. 921\)](#)
- [Calidad de la clave de certificado de \(p. 922\)](#)
- [Calidad de clave de certificado de CA \(p. 924\)](#)
- [Función de Cognito no autenticada excesivamente permisiva \(p. 925\)](#)
- [Función de Cognito autenticada excesivamente permisiva \(p. 930\)](#)
- [AWS IoT Políticas de demasiado permisivas \(p. 937\)](#)
- [Alias de rol excesivamente permisivo \(p. 941\)](#)

- El alias de rol permite el acceso a servicios no utilizados (p. 942)
- El certificado de CA caducado (p. 943)
- ID de cliente MQTT conflictivos (p. 943)
- El certificado de dispositivo caducado (p. 944)
- El certificado de dispositivo revocado sigue activo (p. 945)
- Registro deshabilitado (p. 946)

Certificado de CA revocado pero los certificados de dispositivo siguen activos

Se revocó un certificado de CA pero sigue activo en AWS IoT.

Esta comprobación aparece como `MOREVOKED_CA_CERTIFICATE_STILL_ACTIVE_CHECK` en la CLI y la API.

Gravedad: Critical

Detalles

Un certificado de CA está marcado como revocado en la lista de revocación de certificados mantenida por la entidad emisora, pero sigue marcado como "ACTIVE" o "PENDING_TRANSFER" en AWS IoT.

Se devuelven los siguientes códigos de motivo cuando esta comprobación encuentra un certificado de CA no conforme:

- `CERTIFICATE_REVOKED_BY_ISSUER`

¿Por qué importa?

Un certificado de CA revocado no debe utilizarse nunca más para firmar certificados de dispositivos. Puede que se haya revocado porque ha sufrido un ataque. Los dispositivos agregados recientemente con certificados firmados con este certificado de CA pueden constituir una amenaza para la seguridad.

Cómo solucionarlo

1. Utilice [UpdateCACertificate](#) para marcar el certificado de CA como INACTIVO en AWS IoT. También puede utilizar acciones de mitigación para:
 - Aplicar la acción de mitigación `UPDATE_CA_CERTIFICATE` en los resultados de la auditoría para realizar este cambio.
 - Aplicar la acción `PUBLISH_FINDINGS_TO_SNS` para implementar una respuesta personalizada al mensaje de Amazon SNS.

Para obtener más información, consulte [Acciones de mitigación \(p. 1036\)](#).

2. Revise la actividad de registro de certificados de dispositivo realizada después de que se revocara el certificado de CA y considere la posibilidad de revocar los certificados de dispositivo que se hayan podido emitir durante este período. Puede utilizar [ListCertificatesByCA](#) para obtener una lista de los certificados de dispositivo firmados con el certificado de CA y [UpdateCertificate](#) para revocar un certificado de dispositivo.

Certificado de dispositivo compartido

Varias conexiones simultáneas usan el mismo certificado X.509 para autenticarse con AWS IoT.

Esta comprobación aparece como `DEVICE_CERTIFICATE_SHARED_CHECK` en la CLI y la API.

Gravedad: Critical

Detalles

Cuando se realiza como parte de una auditoría bajo demanda, esta comprobación analiza los certificados y los ID de cliente utilizados por los dispositivos para conectarse durante los 31 días anteriores al inicio de la auditoría hasta 2 horas anteriores al inicio de la auditoría hasta 2 horas anteriores a la ejecución de la comprobación. Para las auditorías programadas, esta comprobación analiza los datos desde 2 horas anteriores a la última vez que se realizó la auditoría hasta 2 horas anteriores al momento en que comenzó esta instancia de la auditoría. Si ha tomado medidas para mitigar esta condición durante el periodo de la comprobación, observe cuándo se realizaron las conexiones simultáneas para determinar si el problema persiste.

Se devuelven los siguientes códigos de motivo cuando esta comprobación encuentra un certificado no conforme:

- `CERTIFICATE_SHARED_BY_MULTIPLE_DEVICES`

Además, los resultados devueltos con esta comprobación incluyen el ID del certificado compartido, los ID de los clientes que usan el certificado para conectarse y los tiempos de conexión/desconexión. Se muestran primero los resultados más recientes.

¿Por qué importa?

Cada dispositivo debe tener un certificado único para autenticarse con AWS IoT. Cuando varios dispositivos utilizan el mismo certificado, esto podría indicar que un dispositivo ha sufrido un ataque. Puede que su identidad haya sido clonada para realizar nuevos ataques en el sistema.

Cómo solucionarlo

Compruebe que el certificado del dispositivo no ha sufrido un ataque. Si ha sido atacado, siga las prácticas recomendadas de seguridad para mitigar la situación.

Si utiliza el mismo certificado en varios dispositivos, es posible que desee:

1. Aprovisionar nuevo certificados únicos y asociarlos a cada dispositivo.
2. Verificar que los nuevos certificados sean válidos y que los dispositivos puedan usarlos para conectarse.
3. Utilice [UpdateCertificate](#) para marcar el certificado antiguo como REVOCADO en AWS IoT. También puede utilizar acciones de mitigación para realizar las siguientes acciones:
 - Aplicar la acción de mitigación `UPDATE_DEVICE_CERTIFICATE` en los resultados de la auditoría para realizar este cambio.
 - Aplicar la acción de mitigación `ADD_THINGS_TO_THING_GROUP` para añadir el dispositivo a un grupo en el que puede tomar medidas.
 - Aplicar la acción de mitigación `PUBLISH_FINDINGS_TO_SNS` si desea implementar una respuesta personalizada al mensaje de Amazon SNS.

Para obtener más información, consulte [Acciones de mitigación \(p. 1036\)](#).

4. Desvincular el certificado antiguo de cada uno de los dispositivos.

Calidad de la clave de certificado de

Los clientes de AWS IoT suelen confiar en la autenticación mutua de TLS mediante certificados X.509 para autenticarse en el agente de mensajes de AWS IoT. Estos certificados y los certificados de la entidad

de certificación se deben registrar en la cuenta de AWS IoT para poder utilizarlos. AWS IoT realiza comprobaciones de verificación básicas en estos certificados cuando se registran. Estas comprobaciones incluyen:

- Deben tener un formato válido.
- Deben estar firmados por una autoridad de certificación registrada.
- Deben estar dentro del periodo de validez (es decir, no han caducado).
- Los tamaños de la clave criptográfica deben cumplir con un tamaño mínimo requerido (para las claves RSA deben ser de 2048 bits o más).

Esta comprobación de auditoría proporciona las siguientes pruebas adicionales de la calidad de la clave criptográfica:

- CVE-2008-0166: comprueba si la clave se generó usando OpenSSL 0.9.8c-1 hasta versiones anteriores a 0.9.8g-9 en un sistema operativo basado en Debian. Esas versiones de OpenSSL utilizan un generador de números aleatorios que genera números predecibles, lo que permite a los atacantes remotos realizar ataques de adivinación de fuerza bruta contra claves criptográficas.
- CVE-2017-15361: comprueba si la clave se generó mediante la biblioteca Infineon RSA 1.02.013 en el firmware Infineon Trusted Platform Module (TPM), como las versiones anteriores a 000000000422 — 4.34, anteriores a 00000000062b — 6.43 y anteriores a 000000008521 — 133.33. Esta biblioteca trata de forma inadecuada la generación de claves RSA, lo que permite a los atacantes frustrar algunos mecanismos de protección criptográfica a través de ataques dirigidos. Algunos ejemplos de tecnologías afectadas son BitLocker con TPM 1.2, generación de claves PGP de YubiKey 4 (anterior a 4.3.5) y la función de cifrado de datos de usuario en caché en el sistema operativo Chrome.

AWS IoT Device Defender registra los certificados como no conformes si no superan estas comprobaciones.

Esta comprobación aparece como DEVICE_CERTIFICATE_KEY_QUALITY_CHECK en la CLI y la API.

Gravedad: Critical

Detalles

Esta comprobación se aplica a los certificados de dispositivo ACTIVE o PENDING_TRANSFER.

Se devuelven los siguientes códigos de motivo cuando esta comprobación encuentra un certificado no conforme:

- CERTIFICATE_KEY_VULNERABILITY_CVE-2017-15361
- CERTIFICATE_KEY_VULNERABILITY_CVE-2008-0166

¿Por qué importa?

Cuando un dispositivo utiliza un certificado vulnerable, los atacantes pueden atacar más fácilmente ese dispositivo.

Cómo solucionarlo

Actualice los certificados de su dispositivo para reemplazar aquellos con vulnerabilidades conocidas.

Si está utilizando el mismo certificado en varios dispositivos, es posible que desee:

1. Aprovisionar nuevo certificados únicos y asociarlos a cada dispositivo.
2. Verificar que los nuevos certificados sean válidos y que los dispositivos puedan usarlos para conectarse.

3. Utilice [UpdateCertificate](#) para marcar el certificado antiguo como REVOCADO en AWS IoT. También puede utilizar acciones de mitigación para:
 - Aplicar la acción de mitigación UPDATE_DEVICE_CERTIFICATE en los resultados de la auditoría para realizar este cambio.
 - Aplicar la acción de mitigación ADD_THINGS_TO_THING_GROUP para añadir el dispositivo a un grupo en el que puede tomar medidas.
 - Aplicar el PUBLISH_FINDINGS_TO_SNS Acción de mitigación si desea implementar una respuesta personalizada al mensaje de Amazon SNS.

Para obtener más información, consulte [Acciones de mitigación \(p. 1036\)](#).

4. Desvincular el certificado antiguo de cada uno de los dispositivos.

Calidad de clave de certificado de CA

Los clientes de AWS IoT suelen confiar en la autenticación mutua de TLS mediante certificados X.509 para autenticarse en el agente de mensajes de AWS IoT. Estos certificados y los certificados de la entidad de certificación deben registrarse en la cuenta de AWS IoT para poder utilizarlos. AWS IoT realiza comprobaciones de verificación básicas en estos certificados cuando se registran, incluidas las siguientes:

- Los certificados tienen un formato válido.
- Los certificados están dentro de su período de validez (es decir, no han caducado).
- Los tamaños de la clave criptográfica cumplen con un tamaño mínimo requerido (para las claves RSA deben ser de 2048 bits o más).

Esta comprobación de auditoría proporciona las siguientes pruebas adicionales de la calidad de la clave criptográfica:

- CVE-2008-0166: comprueba si la clave se generó usando OpenSSL 0.9.8c-1 hasta versiones anteriores a 0.9.8g-9 en un sistema operativo basado en Debian. Esas versiones de OpenSSL utilizan un generador de números aleatorios que genera números predecibles, lo que permite a los atacantes remotos realizar ataques de adivinación de fuerza bruta contra claves criptográficas.
- CVE-2017-15361: comprueba si la clave se generó mediante la biblioteca Infineon RSA 1.02.013 en el firmware Infineon Trusted Platform Module (TPM), como las versiones anteriores a 000000000422 — 4.34, anteriores a 00000000062b — 6.43 y anteriores a 000000008521 — 133.33. Esta biblioteca trata de forma inadecuada la generación de claves RSA, lo que permite a los atacantes frustrar algunos mecanismos de protección criptográfica a través de ataques dirigidos. Algunos ejemplos de tecnologías afectadas son BitLocker con TPM 1.2, generación de claves PGP de YubiKey 4 (anterior a 4.3.5) y la función de cifrado de datos de usuario en caché en el sistema operativo Chrome.

AWS IoT Device Defender registra los certificados como no conformes si no superan estas comprobaciones.

Esta comprobación aparece como `CERTIFICATE_KEY_QUALITY_CHECK` en la CLI y la API.

Gravedad: Critical

Detalles

Esta comprobación se aplica a los certificados de CA ACTIVE o PENDING_TRANSFER.

Se devuelven los siguientes códigos de motivo cuando esta comprobación encuentra un certificado no conforme:

- CERTIFICATE_KEY_VULNERABILITY_CVE-2017-15361

- CERTIFICATE_KEY_VULNERABILITY_CVE-2008-0166

¿Por qué importa?

Los dispositivos agregados recientemente firmados con este certificado de CA pueden constituir una amenaza para la seguridad.

Cómo solucionarlo

1. Utilice [UpdateCACertificate](#) para marcar el certificado de CA como INACTIVO en AWS IoT. También puede utilizar acciones de mitigación para:
 - Aplicar la acción de mitigación UPDATE_CA_CERTIFICATE en los resultados de la auditoría para realizar este cambio.
 - Aplicar el PUBLISH_FINDINGS_TO_SNS Acción de mitigación si desea implementar una respuesta personalizada al mensaje de Amazon SNS.

Para obtener más información, consulte [Acciones de mitigación \(p. 1036\)](#).

2. Revise la actividad de registro de certificados de dispositivo realizada después de que se revocara el certificado de CA y considere la posibilidad de revocar los certificados de dispositivo que se hayan podido emitir durante este período. (Utilice [ListCertificatesByCA](#) para obtener una lista de los certificados de dispositivos firmados con el certificado de CA y [UpdateCertificate](#) para revocar un certificado de dispositivos).

Función de Cognito no autenticada excesivamente permisiva

Una política asociada a un rol de grupo de identidades de Amazon Cognito no autenticado se considera excesivamente permisivo porque otorga permiso para realizar cualquiera de las siguientes operaciones:AWS IoT Acciones:

- Administrar o modificar objetos
- Leer datos administrativos de objetos
- Administrar datos o recursos relacionados con elementos que no sean objetos

O bien porque otorga permiso para realizar las siguientes acciones de AWS IoT en un amplio conjunto de dispositivos:

- Utilizar MQTT para conectar, publicar o suscribirse a temas reservados (incluidos los datos de ejecución de sombras o de trabajos)
- Utilizar comandos de la API para leer o modificar los datos de ejecución de sombras o de trabajos

En general, los dispositivos que se conectan usando un rol de grupo de identidades de Amazon Cognito no autenticado deben tener solo permisos limitados para publicar y suscribirse a temas de MQTT específicos o usar los comandos de la API para leer y modificar datos específicos de objetos relacionados con los datos de ejecución de sombras o de trabajos.

Esta comprobación aparece como UNAUTHENTICATED_COGNITO_ROLE_OVERLY_PERMISSIVE_CHECK en la CLI y la API.

Gravedad: Critical

Detalles

Para esta comprobación,AWS IoT Device Defenderaudita todos los grupos de identidades de Amazon Cognito que se han utilizado para conectarse al AWS IoT Agente de mensajes durante los 31 días anteriores

a la ejecución de la auditoría. En la auditoría se incluyen todos los grupos de identidades de Amazon Cognito a partir de los cuales se ha conectado una identidad de Amazon Cognito autenticada o no autenticada.

Se devuelven los siguientes códigos de motivo cuando esta comprobación encuentra un rol de grupo de identidades de Amazon Cognito no autenticado no conforme:

- `ALLOWS_ACCESS_TO_IOT_ADMIN_ACTIONS`
- `ALLOWS_BROAD_ACCESS_TO_IOT_DATA_PLANE_ACTIONS`

¿Por qué importa?

Debido a que las identidades no autenticadas nunca las autentica el usuario, suponen un riesgo mucho mayor que las identidades de Amazon Cognito autenticadas. Si se ha puesto en riesgo una identidad no autenticada, se podrían usar acciones administrativas para modificar la configuración de la cuenta, eliminar recursos u obtener acceso a información confidencial. O bien, con un amplio acceso a la configuración del dispositivo, se podría obtener acceso a sombras y trabajos de todos los dispositivos de su cuenta o modificarlos. Un usuario invitado podría usar los permisos para poner en riesgo toda su flota o lanzar un ataque DDOS con mensajes.

Cómo solucionarlo

Una política asociada a un rol de grupo de identidades de Amazon Cognito no autenticado debería otorgar solo los permisos necesarios para que un dispositivo haga su trabajo. Recomendamos los siguientes pasos:

1. Crear un nuevo rol conforme.
2. Crear un nuevo grupo de identidades de Amazon Cognito y asociarlo al rol conforme.
3. Verificar que sus identidades puedan obtener acceso a AWS IoT con el nuevo grupo.
4. Una vez que se haya completado la verificación, asociar el nuevo rol conforme al grupo de identidades de Amazon Cognito marcado como no conforme.

También puede utilizar acciones de mitigación para:

- Aplicar el `PUBLISH_FINDINGS_TO_SNS` Acción de mitigación para implementar una respuesta personalizada al mensaje de Amazon SNS.

Para obtener más información, consulte [Acciones de mitigación \(p. 1036\)](#).

Administrar o modificar objetos

Las siguientes acciones de la API de AWS IoT se utilizan para administrar o modificar objetos. No se debe otorgar permiso para realizar estas acciones a los dispositivos que se conectan a través de un grupo de identidades de Amazon Cognito no autenticado.

- `AddThingToThingGroup`
- `AttachThingPrincipal`
- `CreateThing`
- `DeleteThing`
- `DetachThingPrincipal`
- `ListThings`
- `ListThingsInThingGroup`
- `RegisterThing`

- RemoveThingFromThingGroup
- UpdateThing
- UpdateThingGroupsForThing

Cualquier rol que otorgue permiso para realizar estas acciones, incluso en un solo recurso, se considera no conforme.

Leer datos administrativos de objetos

Las siguientes acciones de la API de AWS IoT se utilizan para leer o modificar datos de objetos. Los dispositivos que se conectan a través de un grupo de identidades de Amazon Cognito no autenticado no deben recibir permiso para realizar estas acciones.

- DescribeThing
- ListJobExecutionsForThing
- ListThingGroupsForThing
- ListThingPrincipals

Example

- no conforme:

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

Esto permite que el dispositivo realice las acciones especificadas incluso aunque se otorgue solo para un objeto.

Administrar elementos que no sean objetos

Los dispositivos que se conectan a través de un grupo de identidades de Amazon Cognito no autenticado no deben recibir permiso para realizarlas. AWS IoT Acciones de API distintas de las descritas en estas secciones. Puede administrar su cuenta con una aplicación que se conecta a través de un grupo de identidades de Amazon Cognito no autenticado mediante la creación de un grupo de identidades independiente no utilizado por los dispositivos.

Suscribirse/publicar en temas de MQTT

Los mensajes MQTT se envían a través del agente de mensajes de AWS IoT y los dispositivos los usan para realizar muchas acciones, incluido el acceso y la modificación del estado de la sombra y el estado

de ejecución del trabajo. Una política que otorga permiso a un dispositivo para conectarse, publicar o suscribirse a mensajes de MQTT debe restringir estas acciones a recursos específicos de la siguiente manera:

Conectarse

- no conforme:

```
arn:aws:iot:region:account-id:client/*
```

El carácter comodín * permite que cualquier dispositivo se conecte a AWS IoT.

```
arn:aws:iot:region:account-id:client/${iot:ClientId}
```

A menos que `iot:Connection.Thing.IsAttached` se establezca en `true` en la claves de condición, es equivalente al carácter comodín * del ejemplo anterior.

- conforme:

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

La especificación de recursos contiene una variable que coincide con el nombre del dispositivo que se utiliza para conectarse. La instrucción de condición restringe aún más el permiso al comprobar que el certificado utilizado por el cliente de MQTT coincide con el que está asociado al objeto con el nombre utilizado.

Publicación

- no conforme:

```
arn:aws:iot:region:account-id:topic/$aws/things/*/shadow/update
```

Esto permite que el dispositivo actualice la sombra de cualquier dispositivo (* = todos los dispositivos).

```
arn:aws:iot:region:account-id:topic/$aws/things/*
```

Esto permite que el dispositivo lea, actualice o elimine la sombra de cualquier dispositivo.

- conforme:

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [ "iot:Publish" ],  
            "Topic": "$aws/things/${iot:ThingName}/shadow/*"  
        }  
    ]  
}
```

```
        "Action": [ "iot:Publish" ],
        "Resource": [
            "arn:aws:iot:region:account-id:topic/$aws/things/
${iot:Connection.Thing.ThingName}/shadow/*"
        ],
    }
}
```

La especificación del recurso contiene un comodín, pero solo coincide con cualquier tema relacionado con la sombra para el dispositivo cuyo nombre de objeto se utilice para conectarse.

Suscribirse

- no conforme:

```
arn:aws:iot:region:account-id:topicfilter/$aws/things/*
```

Esto permite que el dispositivo se suscriba a temas de sombra o de trabajo reservados para todos los dispositivos.

```
arn:aws:iot:region:account-id:topicfilter/$aws/things/*
```

Lo mismo que el ejemplo anterior, pero usando el comodín #.

```
arn:aws:iot:region:account-id:topicfilter/$aws/things/#/shadow/update
```

Esto permite que el dispositivo vea las actualizaciones de la sombra en cualquier dispositivo (+ = todos los dispositivos).

- conforme:

```
{
    "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/*"
            ],
        }
    ]
}
```

Las especificaciones de recursos contienen caracteres comodín pero solo coinciden con cualquier tema relacionado con la sombra y cualquier tema relacionado con el trabajo para el dispositivo cuyo nombre de objeto se use para conectarse.

Recibir

- conforme:

```
arn:aws:iot:region:account-id:topicfilter/$aws/things/*
```

Esto se permite porque el dispositivo solo puede recibir mensajes de temas en los que tiene permiso para suscribirse.

Leer/modificar los datos de trabajo o sombras

Una política que concede permiso a un dispositivo para realizar una acción de la API para obtener acceso a datos de ejecución de sombras o trabajos o modificarlos debe restringir estas acciones a recursos específicos. Las acciones de la API son las siguientes:

- DeleteThingShadow
- GetThingShadow
- UpdateThingShadow
- DescribeJobExecution
- GetPendingJobExecutions
- StartNextPendingJobExecution
- UpdateJobExecution

Example

- no conforme:

```
arn:aws:iot:region:account-id:thing/*
```

Esto permite al dispositivo realizar la acción especificada en cualquier objeto.

- conforme:

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

Esto permite que el dispositivo realice las acciones especificadas en solo dos objetos.

Función de Cognito autenticada excesivamente permisiva

Una política asociada a un rol de grupo de identidades de Amazon Cognito autenticado se considera excesivamente permisivo porque otorga permiso para realizar lo siguiente: AWS IoT acciones:

- Administrar o modificar objetos
- Administrar datos o recursos relacionados con elementos que no sean objetos

O bien porque otorga permiso para realizar las siguientes acciones de AWS IoT en un amplio conjunto de dispositivos:

- Leer datos administrativos de objetos
- Utilizar MQTT para conectar/publicar/suscribirse a temas reservados (incluidos los datos de ejecución de sombras o de trabajos)
- Utilizar comandos de la API para leer o modificar los datos de ejecución de sombras o de trabajos

En general, los dispositivos que se conectan usando un rol de grupo de identidades de Amazon Cognito autenticado solo deben tener permisos limitados para leer datos administrativos específicos de los objetos, publicar y suscribirse a temas de MQTT específicos o usar los comandos de la API para leer y modificar datos específicos de objetos relacionados con los objetos sombras o de trabajos. datos de ejecución.

Esta comprobación aparece como AUTHENTICATED_COGNITO_ROLE_OVERLY_PERMISSIVE_CHECK en la CLI y la API.

Gravedad: Critical

Detalles

Para esta comprobación, AWS IoT Device Defender audita todos los grupos de identidades de Amazon Cognito que se han utilizado para conectarse al AWS IoT Agente de mensajes durante los 31 días anteriores a la ejecución de la auditoría. En la auditoría se incluyen todos los grupos de identidades de Amazon Cognito a partir de los cuales se ha conectado una identidad de Amazon Cognito autenticada o no autenticada.

Se devuelven los siguientes códigos de motivo cuando esta comprobación encuentra un rol de grupo de identidades de Amazon Cognito autenticado no conforme:

- 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 qué importa?

Si se ha puesto en riesgo una identidad autenticada, se podrían usar acciones administrativas para modificar la configuración de la cuenta, eliminar recursos u obtener acceso a información confidencial.

Cómo solucionarlo

Una política asociada a un rol de grupo de identidades de Amazon Cognito autenticado debería otorgar solo los permisos necesarios para que un dispositivo haga su trabajo. Recomendamos los siguientes pasos:

1. Crear un nuevo rol conforme.
2. Crear un nuevo grupo de identidades de Amazon Cognito y asociarlo al rol conforme.
3. Verificar que sus identidades puedan obtener acceso a AWS IoT con el nuevo grupo.
4. Una vez que se haya completado la verificación, asocie el rol al grupo de identidades de Amazon Cognito marcado como no conforme.

También puede utilizar acciones de mitigación para:

- Aplicar el PUBLISH_FINDINGS_TO_SNS Acción de mitigación para implementar una respuesta personalizada en respuesta al mensaje de Amazon SNS.

Para obtener más información, consulte [Acciones de mitigación \(p. 1036\)](#).

Administrar o modificar objetos

Los siguientes ejemplos de AWS IoT Las acciones de la API se utilizan para administrar o modificar objetos, por lo que no se debe otorgar permiso para realizarlas a los dispositivos que se conectan a través de un grupo de identidades de Amazon Cognito autenticado:

- `AddThingToThingGroup`
- `AttachThingPrincipal`
- `CreateThing`
- `DeleteThing`
- `DetachThingPrincipal`
- `ListThings`
- `ListThingsInThingGroup`
- `RegisterThing`
- `RemoveThingFromThingGroup`
- `UpdateThing`
- `UpdateThingGroupsForThing`

Cualquier rol que otorgue permiso para realizar estas acciones, incluso en un solo recurso, se considera no conforme.

Administrar elementos que no sean objetos

Los dispositivos que se conectan a través de un grupo de identidades de Amazon Cognito autenticado no deben recibir permiso para realizarlas. AWS IoT Acciones de API distintas de las descritas en estas secciones. Para administrar su cuenta con una aplicación que se conecta a través de un grupo de identidades de Amazon Cognito autenticado, cree un grupo de identidades independiente no utilizado por los dispositivos.

Leer datos administrativos de objetos

Los siguientes ejemplos de AWS IoT Las acciones de la API se utilizan para leer datos de objetos, por lo que a los dispositivos que se conectan a través de un grupo de identidades de Amazon Cognito autenticado no se les debe dar permiso para realizarlas solamente en un conjunto limitado de objetos:

- `DescribeThing`
 - `ListJobExecutionsForThing`
 - `ListThingGroupsForThing`
 - `ListThingPrincipals`
-
- no conforme:

```
arn:aws:iot:region:account-id:thing/*
```

Esto permite al dispositivo realizar la acción especificada en cualquier objeto.

- conforme:

```
{  
    "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"
    ]
}
```

Esto permite al dispositivo realizar las acciones especificadas solo en un objeto.

- conforme:

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

Esto es conforme porque, aunque el recurso se especifica con un carácter comodín (*), va precedido de una cadena específica y esta limita el acceso al conjunto de objetos que tienen el prefijo concreto.

- no conforme:

```
arn:aws:iot:region:account-id:thing/*
```

Esto permite al dispositivo realizar la acción especificada en cualquier objeto.

- conforme:

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

```
    ]
}
```

Esto permite al dispositivo realizar las acciones especificadas solo en un objeto.

- conforme:

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

Esto es conforme porque, aunque el recurso se especifica con un carácter comodín (*), va precedido de una cadena específica y esta limita el acceso al conjunto de objetos que tienen el prefijo concreto.

Suscribirse/publicar en temas de MQTT

Los mensajes de MQTT se envían a través del agente de mensajes de AWS IoT y los dispositivos los usan para realizar muchas acciones diferentes, incluido el acceso y la modificación del estado de la sombra y el estado de ejecución del trabajo. Una política que otorga permiso a un dispositivo para conectarse, publicar o suscribirse a mensajes de MQTT debe restringir estas acciones a recursos específicos de la siguiente manera:

Conectar

- no conforme:

```
arn:aws:iot:region:account-id:client/*
```

El carácter comodín * permite que cualquier dispositivo se conecte a AWS IoT.

```
arn:aws:iot:region:account-id:client/${iot:ClientId}
```

A menos que `iot:Connection.Thing.IsAttached` se establezca en true en la claves de condición, es equivalente al carácter comodín * del ejemplo anterior.

- conforme:

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

La especificación del recurso contiene una variable que coincide con el nombre del dispositivo utilizado para conectarse, y la instrucción de condición restringe aún más el permiso verificando que el certificado utilizado por el cliente MQTT coincide con el que está asociado al objeto con el nombre utilizado.

Publicación

- no conforme:

```
arn:aws:iot:region:account-id:topic/$aws/things/*shadow/update
```

Esto permite que el dispositivo actualice la sombra de cualquier dispositivo (* = todos los dispositivos).

```
arn:aws:iot:region:account-id:topic/$aws/things/*
```

Esto permite que el dispositivo lea/actualice/elimine la sombra de cualquier dispositivo.

- conforme:

```
{
    "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/*"
            ],
        }
    ]
}
```

La especificación del recurso contiene un comodín, pero solo coincide con cualquier tema relacionado con la sombra para el dispositivo cuyo nombre de objeto se utilice para conectarse.

Suscribirse

- no conforme:

```
arn:aws:iot:region:account-id:topicfilter/$aws/things/*
```

Esto permite que el dispositivo se suscriba a temas de sombra o de trabajo reservados para todos los dispositivos.

```
arn:aws:iot:region:account-id:topicfilter/$aws/things/#
```

Lo mismo que el ejemplo anterior, pero usando el comodín #.

```
arn:aws:iot:region:account-id:topicfilter/$aws/things/+shadow/update
```

Esto permite que el dispositivo vea las actualizaciones de la sombra en cualquier dispositivo (+ = todos los dispositivos).

- conforme:

```
{  
    "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/*"  
            ],  
        }  
    ]  
}
```

Las especificaciones de recursos contienen caracteres comodín pero solo coinciden con cualquier tema relacionado con la sombra y cualquier tema relacionado con el trabajo para el dispositivo cuyo nombre de objeto se use para conectarse.

Recibir

- conforme:

```
arn:aws:iot:region:account-id:topicfilter/$aws/things/*
```

Esto es conforme porque el dispositivo solo puede recibir mensajes de temas en los que tiene permiso para suscribirse.

Leer o modificar datos de trabajo o sombras

Una política que concede permiso a un dispositivo para realizar una acción de la API para obtener acceso a datos de ejecución de sombras o trabajos o modificarlos debe restringir estas acciones a recursos específicos. Las acciones de la API son las siguientes:

- DeleteThingShadow
- GetThingShadow
- UpdateThingShadow
- DescribeJobExecution
- GetPendingJobExecutions
- StartNextPendingJobExecution
- UpdateJobExecution

Ejemplos

- no conforme:

```
arn:aws:iot:region:account-id:thing/*
```

Esto permite al dispositivo realizar la acción especificada en cualquier objeto.

- conforme:

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

Esto permite al dispositivo realizar las acciones especificadas solo en dos objetos.

AWS IoT políticas de demasiado permisivas

Una política de AWS IoT otorga permisos que son demasiado amplios o no están sujetos a restricciones. Otorga permiso para enviar o recibir mensajes de MQTT para un amplio conjunto de dispositivos, u otorga permiso para obtener acceso a los datos de ejecución de sombras y de trabajos o modificarlos para un amplio conjunto de dispositivos.

En general, una política para un dispositivo debería otorgar acceso a recursos asociados con ese dispositivo y sin otros dispositivos o con muy pocos. Con algunas excepciones, el uso de un carácter comodín (por ejemplo, "") para especificar recursos en dicha política se considera demasiado amplio o sin restricciones.

Esta comprobación aparece como `IOT_POLICY_OVERLY_PERMISSIVE_CHECK` en la CLI y la API.

Gravedad: Critical

Detalles

Se devuelve el siguiente código de motivo cuando esta comprobación encuentra una política de AWS IoT no conforme:

- `ALLOWS_BROAD_ACCESS_TO_IOT_DATA_PLANE_ACTIONS`

¿Por qué importa?

Un certificado, una identidad de Amazon Cognito o un grupo de objetos con una política excesivamente permisiva puede, en caso de ponerse en riesgo, afectar a la seguridad de toda su cuenta. Un atacante podría usar un acceso tan amplio para leer o modificar sombras, trabajos o ejecuciones de trabajos para todos sus dispositivos. O un atacante podría usar un certificado atacado para conectar dispositivos maliciosos o lanzar un ataque DDOS en su red.

Cómo solucionarlo

Siga estos pasos para corregir las políticas no conformes asociadas a objetos, grupos de objetos u otras entidades:

1. Utilice [CreatePolicyVersion](#) para crear una nueva versión conforme de la política. Establezca la marca `setAsDefault` en true. (Esto hace que esta nueva versión funcione para todas las entidades que utilizan la política).
2. Utilice [ListTargetsForPolicy](#) para obtener una lista de destinos (certificados o grupos de objetos) a los que la política está asociada y determinar qué dispositivos están incluidos en los grupos o cuáles utilizan los certificados para conectarse.
3. Verifique que todos los dispositivos asociados puedan conectarse a AWS IoT. Si un dispositivo no puede conectarse, utilice [SetPolicyVersion](#) para devolver la política predeterminada a la versión anterior, revisar la política e intentarlo de nuevo.

Puede utilizar acciones de mitigación para:

- Aplicar la acción de mitigación `REPLACE_DEFAULT_POLICY_VERSION` en los resultados de la auditoría para realizar este cambio.
- Aplicar el `PUBLISH_FINDINGS_TO_SNS` acción de mitigación si desea implementar una respuesta personalizada al mensaje de Amazon SNS.

Para obtener más información, consulte [Acciones de mitigación \(p. 1036\)](#).

Utilice las [variables de política de AWS IoT \(p. 337\)](#) para hacer referencia de forma dinámica a los recursos de AWS IoT en las políticas.

Permisos de MQTT

Los mensajes MQTT se envían a través del agente de mensajes de AWS IoT y los dispositivos los usan para realizar muchas acciones, incluido el acceso y la modificación del estado de la sombra y el estado de ejecución del trabajo. Una política que otorga permiso a un dispositivo para conectarse, publicar o suscribirse a mensajes de MQTT debe restringir estas acciones a recursos específicos de la siguiente manera:

Conectar

- no conforme:

```
arn:aws:iot:region:account-id:client/*
```

El carácter comodín * permite que cualquier dispositivo se conecte a AWS IoT.

```
arn:aws:iot:region:account-id:client/${iot:ClientId}
```

A menos que `iot:Connection.Thing.IsAttached` se establezca en true en la claves de condición, es equivalente al carácter comodín * del ejemplo anterior.

- conforme:

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

La especificación de recursos contiene una variable que coincide con el nombre del dispositivo que se utiliza para conectarse. La instrucción de condición restringe aún más el permiso al comprobar que el certificado utilizado por el cliente de MQTT coincide con el que está asociado al objeto con el nombre utilizado.

Publicación

- no conforme:

```
arn:aws:iot:region:account-id:topic/$aws/things/*shadow/update
```

Esto permite que el dispositivo actualice la sombra de cualquier dispositivo (* = todos los dispositivos).

```
arn:aws:iot:region:account-id:topic/$aws/things/*
```

Esto permite que el dispositivo lea, actualice o elimine la sombra de cualquier dispositivo.

- conforme:

```
{
    "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/*"
            ],
        }
    ]
}
```

La especificación del recurso contiene un comodín, pero solo coincide con cualquier tema relacionado con la sombra para el dispositivo cuyo nombre de objeto se utilice para conectarse.

Suscribirse

- no conforme:

```
arn:aws:iot:region:account-id:topicfilter/$aws/things/*
```

Esto permite que el dispositivo se suscriba a temas de sombra o de trabajo reservados para todos los dispositivos.

```
arn:aws:iot:region:account-id:topicfilter/$aws/things/*
```

Lo mismo que el ejemplo anterior, pero usando el comodín #.

```
arn:aws:iot:region:account-id:topicfilter/$aws/things/+shadow/update
```

Esto permite que el dispositivo vea las actualizaciones de la sombra en cualquier dispositivo (+ = todos los dispositivos).

- conforme:

```
{  
    "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/*"  
            ],  
        }  
    ]  
}
```

Las especificaciones de recursos contienen caracteres comodín pero solo coinciden con cualquier tema relacionado con la sombra y cualquier tema relacionado con el trabajo para el dispositivo cuyo nombre de objeto se use para conectarse.

Recibir

- conforme:

```
arn:aws:iot:region:account-id:topic/$aws/things/*
```

Esto es conforme porque el dispositivo solo puede recibir mensajes de temas en los que tiene permiso para suscribirse.

Permisos de trabajo y sombras

Una política que concede permiso a un dispositivo para realizar una acción de la API para obtener acceso a datos de ejecución de sombras o trabajos o modificarlos debe restringir estas acciones a recursos específicos. Las acciones de la API son las siguientes:

- `DeleteThingShadow`
- `GetThingShadow`
- `UpdateThingShadow`
- `DescribeJobExecution`
- `GetPendingJobExecutions`
- `StartNextPendingJobExecution`
- `UpdateJobExecution`

Ejemplos

- no conforme:

```
arn:aws:iot:region:account-id:thing/*
```

Esto permite al dispositivo realizar la acción especificada en cualquier objeto.

- conforme:

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

Esto permite al dispositivo realizar las acciones especificadas solo en dos objetos.

Alias de rol excesivamente permisivo

AWS IoT proporciona un mecanismo para que los dispositivos conectados se autenticen en AWS IoT utilizando certificados X.509 y luego obtener de corta duración AWS credenciales de un rol de IAM asociado a un AWS IoT alias de rol. Los permisos para estas credenciales se deben reducir mediante políticas de acceso con variables de contexto de autenticación. Si las políticas no están configuradas correctamente, podría quedar expuesto a una escalada de ataque de privilegios. Esta comprobación de auditoría garantiza que las credenciales temporales proporcionadas por los alias de rol de AWS IoT no sean excesivamente permisivas.

Esta comprobación se activa si se encuentra una de las siguientes condiciones:

- La política proporciona permisos administrativos a todos los servicios utilizados en el último año por este alias de rol (por ejemplo, "iot:\"", "dynamodb:\"", "iam:\"", etc.).
- La política proporciona un amplio acceso a acciones de metadatos de objetos, acceso a acciones de AWS IoT restringidas o un amplio acceso a acciones del plano de datos de AWS IoT.
- La política proporciona acceso a servicios de auditoría de seguridad como "iam", "cloudtrail", "guardduty", "inspector" o "trustedadvisor".

Esta comprobación aparece como `IOT_ROLE_ALIAS_OVERLY_PERMISSIVE_CHECK` en la CLI y la API.

Gravedad: Critical

Detalles

Se devuelven los siguientes códigos de motivo cuando esta comprobación encuentra una política de IoT no conforme:

- `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 qué importa?

Al limitar los permisos a aquellos que se requieren para que un dispositivo realice sus operaciones normales, se reducen los riesgos de su cuenta si un dispositivo es víctima de un ataque.

Cómo solucionarlo

Siga estos pasos para corregir las políticas no conformes asociadas a objetos, grupos de objetos u otras entidades:

1. Siga los pasos descritos en [Autorización de llamadas directas a AWS servicios de utilizando AWS IoT Core proveedor de credenciales \(p. 375\)](#) para aplicar una política más restrictiva al alias de rol.

Puede utilizar acciones de mitigación para:

- Aplicar el PUBLISH_FINDINGS_TO_SNS Acción de mitigación si desea implementar una acción personalizada al mensaje de Amazon SNS.

Para obtener más información, consulte [Acciones de mitigación \(p. 1036\)](#).

El alias de rol permite el acceso a servicios no utilizados

AWS IoT alias de rol proporciona un mecanismo para que los dispositivos conectados se autenticen en AWS IoT utilizando certificados X.509 y luego obtener de corta duración AWS credenciales de un rol de IAM asociado a un AWS IoT alias de rol. Los permisos para estas credenciales se deben reducir mediante políticas de acceso con variables de contexto de autenticación. Si las políticas no están configuradas correctamente, podría quedar expuesto a una escalada de ataque de privilegios. Esta comprobación de auditoría garantiza que las credenciales temporales proporcionadas por los alias de rol de AWS IoT no sean excesivamente permisivas.

Esta comprobación se activa si el alias de rol tiene acceso a servicios que no se han utilizado para el dispositivo de AWS IoT en el último año. Por ejemplo, los informes de auditoría si tiene un rol de IAM asociado al alias de rol que solo ha utilizado AWS IoT el año pasado, pero la política asociada a la función también concede permiso para "iam:getRole" y "dynamodb:PutItem".

Esta comprobación aparece como IOT_ROLE_ALIAS_ALLOWS_ACCESS_TO_UNUSED_SERVICES_CHECK en la CLI y la API.

Gravedad: Medio

Detalles

Se devuelven los siguientes códigos de motivo cuando esta comprobación encuentra una política de AWS IoT no conforme:

- ALLOWS_ACCESS_TO_UNUSED_SERVICES

¿Por qué importa?

Al limitar los permisos a los servicios que se requieren para que un dispositivo realice sus operaciones normales, se reducen los riesgos de su cuenta si un dispositivo es víctima de un ataque.

Cómo solucionarlo

Siga estos pasos para corregir las políticas no conformes asociadas a objetos, grupos de objetos u otras entidades:

1. Siga los pasos descritos en [Autorización de llamadas directas a AWS servicios de utilizando AWS IoT Core proveedor de credenciales \(p. 375\)](#) para aplicar una política más restrictiva al alias de rol.

Puede utilizar acciones de mitigación para:

- Aplicar el `PUBLISH_FINDINGS_TO_SNS` acción de mitigación si desea implementar una acción personalizada al mensaje de Amazon SNS.

Para obtener más información, consulte [Acciones de mitigación \(p. 1036\)](#).

El certificado de CA caducado

Un certificado de CA va a caducar dentro de 30 días o ha caducado.

Esta comprobación aparece como `CA_CERTIFICATE_EXPIRING_CHECK` en la CLI y la API.

Gravedad: Medio

Detalles

Esta comprobación se aplica a los certificados de CA `ACTIVE` o `PENDING_TRANSFER`.

Se devuelven los siguientes códigos de motivo cuando esta comprobación encuentra un certificado de CA no conforme:

- `CERTIFICATE_APPROACHING_EXPIRATION`
- `CERTIFICATE_PAST_EXPIRATION`

¿Por qué importa?

Un certificado de CA caducado no debe utilizarse para firmar nuevos certificados de dispositivos.

Cómo solucionarlo

Consulte las prácticas recomendadas de seguridad para saber cómo proceder. Es posible que desee:

1. Registrar un nuevo certificado de CA en AWS IoT.
 2. Verificar que puede firmar certificados de dispositivo con el nuevo certificado de CA.
 3. Utilice [UpdateCACertificate](#) para marcar el antiguo certificado de CA como `INACTIVO` en AWS IoT.
- También puede utilizar acciones de mitigación para realizar las siguientes acciones:
- Aplicar la acción de mitigación `UPDATE_CA_CERTIFICATE` en los resultados de la auditoría para realizar este cambio.
 - Aplicar el `PUBLISH_FINDINGS_TO_SNS` acción de mitigación si desea implementar una respuesta personalizada al mensaje de Amazon SNS.

Para obtener más información, consulte [Acciones de mitigación \(p. 1036\)](#).

ID de cliente MQTT conflictivos

Varios dispositivos se conectan con el mismo ID de cliente.

Esta comprobación aparece como `CONFLICTING_CLIENT_IDS_CHECK` en la CLI y la API.

Gravedad: Alta

Detalles

Se han realizado varias conexiones con el mismo ID de cliente, lo que ha provocado la desconexión de un dispositivo ya conectado. La especificación de MQTT solo permite una conexión activa por ID de cliente, con lo cual, cuando otro dispositivo se conecta con el mismo ID de cliente, se bloquea la conexión del dispositivo anterior.

Cuando se realiza como parte de una auditoría bajo demanda, esta comprobación analiza cómo se usaban los ID de cliente utilizados por los dispositivos para conectarse durante los 31 días anteriores al inicio de la auditoría. Para las auditorías programadas, esta comprobación analiza los datos desde la última vez que se realizó la auditoría hasta el momento en que comenzó esta instancia de la auditoría. Si ha tomado medidas para mitigar esta condición durante el periodo de la comprobación, observe cuándo se realizaron las conexiones/desconexiones para determinar si el problema persiste.

Se devuelven los siguientes códigos de motivo cuando esta comprobación encuentra una falta de conformidad:

- `DUPLICATE_CLIENT_ID_ACROSS_CONNECTIONS`

Además, los resultados devueltos con esta comprobación incluyen el ID de cliente utilizado para conectarse, los identificadores de las entidades principales y los tiempos de desconexión. Se muestran primero los resultados más recientes.

¿Por qué importa?

Los dispositivos con ID en conflicto se ven obligados a volver a conectarse constantemente, lo que puede provocar la pérdida de mensajes o la imposibilidad de que el dispositivo se conecte.

Esto puede indicar que un dispositivo o las credenciales de un dispositivo se han puesto en riesgo y podrían ser parte de un ataque DDoS. También es posible que los dispositivos estén mal configurados en la cuenta o que el dispositivo tenga una mala conexión y se vea obligado a volver a conectarse varias veces por minuto.

Cómo solucionarlo

Registre cada dispositivo como un objeto único en AWS IoT y use el nombre de objeto como ID de cliente para la conexión. O use un UUID como ID de cliente al conectar el dispositivo a través de MQTT. También puede utilizar acciones de mitigación para:

- Aplicar el `PUBLISH_FINDINGS_TO_SNS` Acción de mitigación si desea implementar una respuesta personalizada al mensaje de Amazon SNS.

Para obtener más información, consulte [Acciones de mitigación \(p. 1036\)](#).

El certificado de dispositivo caducado

Un certificado de dispositivo va a caducar dentro de 30 días o ha caducado.

Esta comprobación aparece como `DEVICE_CERTIFICATE_EXPIRING_CHECK` en la CLI y la API.

Gravedad: Medio

Detalles

Esta comprobación se aplica a los certificados de dispositivo `ACTIVE` o `PENDING_TRANSFER`.

Cuando esta comprobación encuentra un certificado de dispositivo no conforme se devuelven los siguientes códigos de motivo:

- CERTIFICATE_APPROACHING_EXPIRATION
- CERTIFICATE_PAST_EXPIRATION

¿Por qué importa?

Un certificado de dispositivo no debe utilizarse una vez que caduque.

Cómo solucionarlo

Consulte las prácticas recomendadas de seguridad para saber cómo proceder. Es posible que desee:

1. Aprovisionar un nuevo certificado y asociarlo al dispositivo.
2. Verificar que el nuevo certificado sea válido y que el dispositivo pueda usarlo para conectarse.
3. Utilice [UpdateCertificate](#) para marcar el certificado antiguo como INACTIVO en AWS IoT. También puede utilizar acciones de mitigación para:
 - Aplicar la acción de mitigación UPDATE_DEVICE_CERTIFICATE en los resultados de la auditoría para realizar este cambio.
 - Aplicar la acción de mitigación ADD_THINGS_TO_THING_GROUP para añadir el dispositivo a un grupo en el que puede tomar medidas.
 - Aplicar el PUBLISH_FINDINGS_TO_SNS Acción de mitigación si desea implementar una respuesta personalizada al mensaje de Amazon SNS.

Para obtener más información, consulte [Acciones de mitigación \(p. 1036\)](#).

4. Desvincular el certificado antiguo del dispositivo. (Consulte [DetachThingPrincipal](#).)

El certificado de dispositivo revocado sigue activo

Un certificado del dispositivo revocado sigue activo.

Esta comprobación aparece como REVOKED_DEVICE_CERTIFICATE_STILL_ACTIVE_CHECK en la CLI y la API.

Gravedad: Medio

Detalles

Un certificado de dispositivo no está en la [lista de revocación de certificados](#) de CA, pero sigue activo en AWS IoT.

Esta comprobación se aplica a los certificados de dispositivo ACTIVE o PENDING_TRANSFER.

Se devuelven los siguientes códigos de motivo cuando esta comprobación encuentra una falta de conformidad:

- CERTIFICATE_REVOKED_BY_ISSUER

¿Por qué importa?

Un certificado de dispositivo generalmente se revoca porque ha sufrido un ataque. Es posible que aún no se haya revocado en AWS IoT debido a un error o un descuido.

Cómo solucionarlo

Compruebe que el certificado del dispositivo no ha sufrido un ataque. Si ha sido atacado, siga las prácticas recomendadas de seguridad para mitigar la situación. Es posible que desee:

1. Aprovisionar un nuevo certificado para el dispositivo.
2. Verificar que el nuevo certificado sea válido y que el dispositivo pueda usarlo para conectarse.
3. Utilice [UpdateCertificate](#) para marcar el certificado antiguo como REVOCADO en AWS IoT. También puede utilizar acciones de mitigación para:
 - Aplicar la acción de mitigación UPDATE_DEVICE_CERTIFICATE en los resultados de la auditoría para realizar este cambio.
 - Aplicar la acción de mitigación ADD_THINGS_TO_THING_GROUP para añadir el dispositivo a un grupo en el que puede tomar medidas.
 - Aplicar el PUBLISH_FINDINGS_TO_SNS Acción de mitigación si desea implementar una respuesta personalizada al mensaje de Amazon SNS.

Para obtener más información, consulte [Acciones de mitigación \(p. 1036\)](#).

4. Desvincular el certificado antiguo del dispositivo. (Consulte [DetachThingPrincipal](#).)

Registro deshabilitado

AWS IoT logs no están activados en Amazon CloudWatch. Verifica el registro de V1 y V2.

Esta comprobación aparece como LOGGING_DISABLED_CHECK en la CLI y la API.

Gravedad: Baja

Detalles

Se devuelven los siguientes códigos de motivo cuando esta comprobación encuentra una falta de conformidad:

- LOGGING_DISABLED

¿Por qué importa?

AWS IoT logs registran en CloudWatch proporcionan visibilidad de los comportamientos en AWS IoT, incluidos errores de autenticación y conexiones y desconexiones inesperadas que podrían indicar que un dispositivo ha sufrido un ataque.

Cómo solucionarlo

Habilitar AWS IoT inicia sesión en CloudWatch. Consulte [Herramientas de monitoreo \(p. 413\)](#). También puede utilizar acciones de mitigación para:

- Aplicar la acción de mitigación ENABLE_IOT_LOGGING en los resultados de la auditoría para realizar este cambio.
- Aplicar el PUBLISH_FINDINGS_TO_SNS Acción de mitigación si desea implementar una respuesta personalizada al mensaje de Amazon SNS.

Para obtener más información, consulte [Acciones de mitigación \(p. 1036\)](#).

Comandos de auditoría

Administrar la configuración de auditorías

Utilice `UpdateAccountAuditConfiguration` para configurar los ajustes de auditoría de su cuenta. Este comando le permite habilitar las comprobaciones que desea que estén disponibles para las auditorías, configurar notificaciones opcionales y configurar permisos.

Compruebe esta configuración con `DescribeAccountAuditConfiguration`.

Utilice `DeleteAccountAuditConfiguration` para eliminar la configuración de auditoría. Esto restaura todos los valores predeterminados y desactiva de manera efectiva las auditorías, ya que todas las comprobaciones están deshabilitadas de manera predeterminada.

UpdateAccountAuditConfiguration

Configura o vuelve a configurar los ajustes de auditoría de Device Defender para esta cuenta. La configuración incluye cómo se envían las notificaciones de auditoría y qué comprobaciones de auditoría se habilitan o deshabilitan.

Sinopsis

```
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 cli-input-json

```
{
  "roleArn": "string",
  "auditNotificationTargetConfigurations": {
    "string": {
      "targetArn": "string",
      "roleArn": "string",
      "enabled": "boolean"
    }
  },
  "auditCheckConfigurations": {
    "string": {
      "enabled": "boolean"
    }
  }
}
```

Campos cli-input-json

Name (Nombre)	Tipo	Descripción
roleArn	string Longitud máx.: 2048; mín.: 20	El ARN del rol que concede permiso a AWS IoT para obtener acceso a la información sobre sus dispositivos, políticas, certificados y otros elementos al realizar una auditoría.

Name (Nombre)	Tipo	Descripción
auditNotificationTargetConfigurations	map	Información sobre los destinos a los que se envían las notificaciones de auditoría.
targetArn	string	El ARN del destino (tema SNS) al que se envían las notificaciones de auditoría.
roleArn	string	El ARN del rol que concede permiso para enviar notificaciones al destino. Longitud máx.: 2048; mín.: 20
enabled	booleano	True si se habilitan las notificaciones al destino.
auditCheckConfigurations	map	<p>Especifica las comprobaciones de auditoría habilitadas y deshabilitadas para esta cuenta. Utilice DescribeAccountAuditConfiguration para ver la lista de todas las comprobaciones, incluidas las habilitadas actualmente.</p> <p>Parte de la recopilación de datos puede comenzar inmediatamente cuando se habilitan determinadas comprobaciones. Cuando se deshabilita una comprobación, se borran todos los datos recopilados hasta el momento relacionados con la comprobación.</p> <p>No puede deshabilitar una comprobación si se utiliza en una auditoría programada. Primero debe eliminar la comprobación de la auditoría programada o eliminar la propia auditoría programada.</p> <p>En la primera llamada a UpdateAccountAuditConfiguration se necesita este parámetro y debe especificar al menos una comprobación habilitada.</p>
enabled	booleano	True si está habilitada la comprobación de auditoría para esta cuenta.

Salida

Ninguno

Errores

InvalidRequestException

El contenido de la solicitud no es válido.

ThrottlingException

El índice supera el límite.

InternalFailureException

Se ha producido un error inesperado.

DescribeAccountAuditConfiguration

Obtiene información sobre la configuración de auditoría de Device Defender para esta cuenta. La configuración incluye cómo se envían las notificaciones de auditoría y qué comprobaciones de auditoría se habilitan o deshabilitan.

Sinopsis

```
aws iot describe-account-audit-configuration \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

Formato cli-input-json

```
{  
}
```

Salida

```
{  
  "roleArn": "string",  
  "auditNotificationTargetConfigurations": {  
    "string": {  
      "targetArn": "string",  
      "roleArn": "string",  
      "enabled": "boolean"  
    }  
  },  
  "auditCheckConfigurations": {  
    "string": {  
      "enabled": "boolean"  
    }  
  }  
}
```

Campos de salida de la CLI

Name (Nombre)	Tipo	Descripción
roleArn	string Longitud máx.: 2048; mín.: 20	El ARN del rol que concede permiso a AWS IoT para obtener acceso a la información sobre sus dispositivos, políticas, certificados y otros elementos al realizar una auditoría.

Name (Nombre)	Tipo	Descripción
		En la primera llamada a <code>UpdateAccountAuditConfiguration</code> , se necesita este parámetro.
auditNotificationTargetConfigurationsMap		Información sobre los destinos a los que se envían las notificaciones de auditoría para esta cuenta.
targetArn	string	El ARN del destino (tema SNS) al que se envían las notificaciones de auditoría.
roleArn	string Longitud máx.: 2048; mín.: 20	El ARN del rol que concede permiso para enviar notificaciones al destino.
enabled	booleano	True si se habilitan las notificaciones al destino.
auditCheckConfigurations	map	Las verificaciones de auditoría habilitadas y deshabilitadas para esta cuenta.
enabled	booleano	True si está habilitada la comprobación de auditoría para esta cuenta.

Errores

`ThrottlingException`

El índice supera el límite.

`InternalFailureException`

Se ha producido un error inesperado.

DeleteAccountAuditConfiguration

Restaura la configuración predeterminada para las auditorías de Device Defender para esta cuenta. Se eliminarán todos los datos de configuración que haya introducido y todas las comprobaciones de auditoría se restablecerán como deshabilitadas.

Sinopsis

```
aws iot delete-account-audit-configuration \
[--delete-scheduled-audits | --no-delete-scheduled-audits] \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

Formato cli-input-json

```
{
  "deleteScheduledAudits": "boolean"
}
```

Campos `cli-input-json`

Name (Nombre)	Tipo	Descripción
deleteScheduledAudits	booleano	Si es true, se eliminan todas las auditorías programadas.

Salida

Ninguno

Errores

`InvalidRequestException`

El contenido de la solicitud no es válido.

`ResourceNotFoundException`

El recurso especificado no existe.

`ThrottlingException`

El índice supera el límite.

`InternalFailureException`

Se ha producido un error inesperado.

Programación de auditorías

Utilice `CreateScheduledAudit` para crear una o varias auditorías programadas. Este comando permite especificar las comprobaciones que desea realizar durante una auditoría y con qué frecuencia deben ejecutarse.

Mantenga un registro de sus auditorías programadas con `ListScheduledAudits` y `DescribeScheduledAudit`.

Cambie una auditoría programada existente con `UpdateScheduledAudit` o elimínela con `DeleteScheduledAudit`.

CreateScheduledAudit

Crea una auditoría programada que se ejecuta en un intervalo de tiempo especificado.

Sinopsis

```
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 `cli-input-json`

```
{
```

```

    "frequency": "string",
    "dayOfMonth": "string",
    "dayOfWeek": "string",
    "targetCheckNames": [
        "string"
    ],
    "tags": [
        {
            "Key": "string",
            "Value": "string"
        }
    ],
    "scheduledAuditName": "string"
}

```

Campos cli-input-json

Name (Nombre)	Tipo	Descripción
frequency	string	Frecuencia con la que se lleva a cabo la auditoría. Puede ser DAILY, WEEKLY, BIWEEKLY o MONTHLY. El sistema determina la hora de inicio real de cada auditoría. enum: DIARIO SEMANAL QUINCENAL MENSUAL
dayOfMonth	string Pattern: ^ ([1-9] [12] [0-9] 3 [01]) \$ ^LAST\$	El día del mes en que tiene lugar la auditoría programada. Puede ser de 1 a 31 o LAST. Este campo es obligatorio si el parámetro frequency se establece en MONTHLY. Si se especifican los días 29-31 y el mes no tiene tantos días, la auditoría se realizará el último (LAST) día del mes.
dayOfWeek	string	El día de la semana en que tiene lugar la auditoría programada. Puede ser SUN, MON, TUE, WED, THU, FRI o SAT. Este campo es obligatorio si el parámetro frequency se establece en WEEKLY o BIWEEKLY. enum: DOM LUN MAR MIÉ JUE VIE SÁB
targetCheckNames	Lista miembro: AuditCheckName	Controles que se realizan durante la auditoría programada. Las comprobaciones deben activarse para su cuenta. (Utilice <code>DescribeAccountAuditConfiguration</code> para ver la lista de todas las comprobaciones, incluidas las habilitadas o

Name (Nombre)	Tipo	Descripción
		UpdateAccountAuditConfiguration para seleccionar las comprobaciones habilitadas).
tags	Lista miembro: Tag Clase de Java: java.util.List	Los metadatos que se pueden utilizar para administrar la auditoría programada.
Clave	string	La clave de la etiqueta.
Valor	string	El valor de la etiqueta.
scheduledAuditName	string Longitud máx.: 128; mín.: 1 Patrón: [a-zA-Z0-9_-]+	El nombre que desea asignar a la auditoría programada. (Máximo de 128 caracteres)

Salida

```
{
  "scheduledAuditArn": "string"
}
```

Campos de salida de la CLI

Name (Nombre)	Tipo	Descripción
scheduledAuditArn	string	El ARN de la auditoría programada.

Errores

`InvalidRequestException`

El contenido de la solicitud no es válido.

`ThrottlingException`

El índice supera el límite.

`InternalFailureException`

Se ha producido un error inesperado.

`LimitExceedededException`

Se ha superado un límite.

ListScheduledAudits

Muestra una lista de las auditorías programadas.

Sinopsis

```
aws iot list-scheduled-audits \
[--next-token <value>] \
[--max-results <value>] \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

Formato cli-input-json

```
{
    "nextToken": "string",
    "maxResults": "integer"
}
```

Campos cli-input-json

Name (Nombre)	Tipo	Descripción
nextToken	string	El token del conjunto siguiente de resultados.
maxResults	entero Rango máx.: 250; mín.: 1	El número máximo de resultados que devolver a la vez. El valor predeterminado es 25.

Salida

```
{
    "scheduledAudits": [
        {
            "scheduledAuditName": "string",
            "scheduledAuditArn": "string",
            "frequency": "string",
            "dayOfMonth": "string",
            "dayOfWeek": "string"
        }
    ],
    "nextToken": "string"
}
```

Campos de salida de la CLI

Name (Nombre)	Tipo	Descripción
scheduledAudits	Lista miembro: ScheduledAuditMetadata Clase de Java: java.util.List	La lista de auditorías programadas.
scheduledAuditName	string Longitud máx.: 128; mín.: 1 Patrón: [a-zA-Z0-9_-]+	El nombre de la auditoría programada.
scheduledAuditArn	string	El ARN de la auditoría programada.

Name (Nombre)	Tipo	Descripción
frequency	string	Frecuencia con la que se lleva a cabo la auditoría. enum: DIARIO SEMANAL QUINCENAL MENSUAL
dayOfMonth	string Pattern: ^ ([1-9] [12] [0-9] 3 [01]) \$ ^LAST\$	El día del mes en que se ejecuta la auditoría programada (si <code>frequency</code> es MONTHLY). Si se especifican los días 29-31 y el mes no tiene tantos días, la auditoría se realizará el último (LAST) día del mes.
dayOfWeek	string	El día de la semana en que se ejecuta la auditoría programada (si <code>frequency</code> es WEEKLY o BIWEEKLY). enum: DOM LUN MAR MIÉ JUE VIE SÁB
nextToken	string	Un token que se puede utilizar para recuperar el siguiente conjunto de resultados o <code>null</code> si no hay resultados adicionales.

Errores

InvalidRequestException

El contenido de la solicitud no es válido.

ThrottlingException

El índice supera el límite.

InternalFailureException

Se ha producido un error inesperado.

DescribeScheduledAudit

Obtiene información acerca de una auditoría programada.

Sinopsis

```
aws iot describe-scheduled-audit \
--scheduled-audit-name <value> \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

Formato cli-input-json

```
{
  "scheduledAuditName": "string"
```

}

Campos `cli-input-json`

Name (Nombre)	Tipo	Descripción
scheduledAuditName	string Longitud máx.: 128; mín.: 1 Patrón: [a-zA-Z0-9_-]+	El nombre de la auditoría programada cuya información desea obtener.

Salida

```
{
  "frequency": "string",
  "dayOfMonth": "string",
  "dayOfWeek": "string",
  "targetCheckNames": [
    "string"
  ],
  "scheduledAuditName": "string",
  "scheduledAuditArn": "string"
}
```

Campos de salida de la CLI

Name (Nombre)	Tipo	Descripción
frequency	string	Frecuencia con la que se lleva a cabo la auditoría. Uno entre DAILY, WEEKLY, BIWEEKLY o MONTHLY. El sistema determina la hora de inicio real de cada auditoría. enum: DIARIO SEMANAL QUINCENAL MENSUAL
dayOfMonth	string Pattern: ^ ([1-9] [12] [0-9] [01]) \$ ^LAST\$	El día del mes en que tiene lugar la auditoría programada. Puede ser de 1 a 31 o LAST. Si se especifican los días 29-31 y el mes no tiene tantos días, la auditoría se realizará el último (LAST) día del mes.
dayOfWeek	string	El día de la semana en que tiene lugar la auditoría programada. Uno entre SUN, MON, TUE, WED, THU, FRI o SAT. enum: DOM LUN MAR MIÉ JUE VIE SÁB
targetCheckNames	Lista miembro: AuditCheckName	Controles que se realizan durante la auditoría programada. Las comprobaciones deben

Name (Nombre)	Tipo	Descripción
		activarse para su cuenta. (Utilice <code>DescribeAccountAuditConfiguration</code> para ver la lista de todas las comprobaciones, incluidas las habilitadas o <code>UpdateAccountAuditConfiguration</code> para seleccionar las comprobaciones habilitadas).
scheduledAuditName	string Longitud máx.: 128; mín.: 1 Patrón: [a-zA-Z0-9_-]+	El nombre de la auditoría programada.
scheduledAuditArn	string	El ARN de la auditoría programada.

Errores

`InvalidRequestException`

El contenido de la solicitud no es válido.

`ResourceNotFoundException`

El recurso especificado no existe.

`ThrottlingException`

El índice supera el límite.

`InternalFailureException`

Se ha producido un error inesperado.

UpdateScheduledAudit

Actualiza una auditoría programada, que incluye las comprobaciones que se realizan y con qué frecuencia se lleva a cabo la auditoría.

Sinopsis

```
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 cli-input-json

```
{
  "frequency": "string",
  "dayOfMonth": "string",
  "dayOfWeek": "string",
  "targetCheckNames": [
```

```

        "string"
],
"scheduledAuditName": "string"
}

```

Campos `cli-input-json`

Name (Nombre)	Tipo	Descripción
frequency	string	Frecuencia con la que se lleva a cabo la auditoría. Puede ser DAILY, WEEKLY, BIWEEKLY o MONTHLY. El sistema determina la hora de inicio real de cada auditoría. enum: DIARIO SEMANAL QUINCENAL MENSUAL
dayOfMonth	string Pattern: ^ ([1-9] [12] [0-9] 3 [01]) \$ ^LAST\$	El día del mes en que tiene lugar la auditoría programada. Puede ser de 1 a 31 o LAST. Este campo es obligatorio si el parámetro <code>frequency</code> se establece en MONTHLY. Si se especifican los días 29-31 y el mes no tiene tantos días, la auditoría se realizará el último (LAST) día del mes.
dayOfWeek	string	El día de la semana en que tiene lugar la auditoría programada. Puede ser SUN, MON, TUE, WED, THU, FRI o SAT. Este campo es obligatorio si el parámetro <code>frequency</code> se establece en WEEKLY o BIWEEKLY. enum: DOM LUN MAR MIÉ JUE VIE SÁB
targetCheckNames	Lista miembro: AuditCheckName	Controles que se realizan durante la auditoría programada. Las comprobaciones deben activarse para su cuenta. (Utilice <code>DescribeAccountAuditConfiguration</code> para ver la lista de todas las comprobaciones, incluidas las habilitadas o <code>UpdateAccountAuditConfiguration</code> para seleccionar las comprobaciones habilitadas).
scheduledAuditName	string Longitud máx.: 128; mín.: 1 Patrón: [a-zA-Z0-9_-]+	El nombre de la auditoría programada. (Máximo de 128 caracteres)

Salida

```
{  
    "scheduledAuditArn": "string"  
}
```

Campos de salida de la CLI

Name (Nombre)	Tipo	Descripción
scheduledAuditArn	string	El ARN de la auditoría programada.

Errores

InvalidRequestException

El contenido de la solicitud no es válido.

ResourceNotFoundException

El recurso especificado no existe.

ThrottlingException

El índice supera el límite.

InternalFailureException

Se ha producido un error inesperado.

DeleteScheduledAudit

Elimina una auditoría programada.

Sinopsis

```
aws iot delete-scheduled-audit \  
  --scheduled-audit-name <value> \  
  [--cli-input-json <value>] \  
  [--generate-cli-skeleton]
```

Formato cli-input-json

```
{  
    "scheduledAuditName": "string"  
}
```

Campos **cli-input-json**

Name (Nombre)	Tipo	Descripción
scheduledAuditName	string	El nombre de la auditoría programada que desea borrar. Longitud máx.: 128; mín.: 1 Patrón: [a-zA-Z0-9_-]+

Salida

Ninguno

Errores

InvalidRequestException

El contenido de la solicitud no es válido.

ResourceNotFoundException

El recurso especificado no existe.

ThrottlingException

El índice supera el límite.

InternalFailureException

Se ha producido un error inesperado.

Ejecutar una auditoría bajo demanda

Utilice `StartOnDemandAuditTask` para especificar las comprobaciones que desea realizar y comenzar una auditoría de inmediato.

StartOnDemandAuditTask

Comienza una auditoría de Device Defender bajo demanda.

Sinopsis

```
aws iot start-on-demand-audit-task \
  --target-check-names <value> \
  [--cli-input-json <value>] \
  [--generate-cli-skeleton]
```

Formato `cli-input-json`

```
{
  "targetCheckNames": [
    "string"
  ]
}
```

Campos `cli-input-json`

Name (Nombre)	Tipo	Descripción
targetCheckNames	Lista miembro: AuditCheckName	Controles que se realizan durante la auditoría. Las comprobaciones que especifique deben estar habilitadas para su cuenta o se producirá una excepción. Utilice <code>DescribeAccountAuditConfiguration</code> para ver la lista de todas las comprobaciones, incluidas las habilitadas o <code>UpdateAccountAuditConfiguration</code> para seleccionar las comprobaciones habilitadas.

Salida

```
{  
    "taskId": "string"  
}
```

Campos de salida de la CLI

Name (Nombre)	Tipo	Descripción
taskId	string Longitud máx.: 40; mín.: 1 Patrón: [a-zA-Z0-9-]+	El ID de la auditoría bajo demanda que comenzó.

Errores

`InvalidRequestException`

El contenido de la solicitud no es válido.

`ThrottlingException`

El índice supera el límite.

`InternalFailureException`

Se ha producido un error inesperado.

`LimitExceedededException`

Se ha superado un límite.

Administrar instancias de auditoría

Utilice `DescribeAuditTask` para obtener información sobre una instancia de auditoría específica. Si ya se ha ejecutado, los resultados incluyen las comprobaciones fallidas y las correctas, aquellas que el sistema no ha podido completar y, si la auditoría aún está en curso, aquellas en las que todavía está trabajando.

Utilice `ListAuditTasks` para buscar las auditorías que se ejecutaron durante un intervalo de tiempo específico.

Utilice `CancelAuditTask` para parar una auditoría en curso.

DescribeAuditTask

Obtiene información acerca de una auditoría de Device Defender.

Sinopsis

```
aws iot describe-audit-task \  
  --task-id <value> \  
  [--cli-input-json <value>] \  
  [--generate-cli-skeleton]
```

Formato cli-input-json

```
{  
    "taskId": "string"
```

}

Campos `cli-input-json`

Name (Nombre)	Tipo	Descripción
taskId	string Longitud máx.: 40; mín.: 1 Patrón: [a-zA-Z0-9-]+	El ID de la auditoría cuya información desea obtener.

Salida

```
{
    "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 salida de la CLI

Name (Nombre)	Tipo	Descripción
taskStatus	string	El estado de la auditoría: uno entre IN_PROGRESS, COMPLETED, FAILED o CANCELED. enum: IN_PROGRESS COMPLETADO ERROR CANCELADO
taskType	string	El tipo de auditoría: ON_DEMAND_AUDIT_TASK o SCHEDULED_AUDIT_TASK. enum: ON_DEMAND_AUDIT_TASK SCHEDULED_AUDIT_TASK

Name (Nombre)	Tipo	Descripción
taskStartTime	timestamp	La hora a la que se inició la auditoría.
taskStatistics	TaskStatistics	Información estadística sobre la auditoría.
totalChecks	entero	El número de comprobaciones de esta auditoría.
inProgressChecks	entero	El número de comprobaciones en curso.
waitingForDataCollectionChecks	entero	El número de comprobaciones en espera de recopilación de datos.
compliantChecks	entero	El número de comprobaciones que encontraron recursos conformes.
nonCompliantChecks	entero	El número de comprobaciones que encontraron recursos no conformes.
failedChecks	entero	El número de comprobaciones.
canceledChecks	entero	El número de comprobaciones no ejecutadas debido a la cancelación de la auditoría.
scheduledAuditName	string Longitud máx.: 128; mín.: 1 Patrón: [a-zA-Z0-9_-]+	El nombre de la auditoría programada (solo si la auditoría era una auditoría programada).
auditDetails	map	Información detallada sobre cada comprobación realizada durante esta auditoría.
checkRunStatus	string	El estado de finalización de esta comprobación, una entre IN_PROGRESS, WAITING_FOR_DATA_COLLECTION, CANCELED, COMPLETED_COMPLIANT, COMPLETED_NON_COMPLIANT o FAILED. enum: IN_PROGRESS WAITING_FOR_DATA_COLLECTION CANCELADO COMPLETED_COMPLIANT COMPLETED_NON_COMPLIANT ERROR

Name (Nombre)	Tipo	Descripción
checkCompliant	booleano	True si se ha completado la comprobación y se ha detectado que todos los recursos son conformes.
totalResourcesCount	long	El número de recursos en los que se ha realizado la comprobación.
nonCompliantResourcesCount	long	El número de recursos que la comprobación ha encontrado que no son conformes.
errorCode	string	El código de cualquier error encontrado al realizar esta comprobación durante esta auditoría. Uno entre INSUFFICIENT_PERMISSIONS o AUDIT_CHECK_DISABLED.
message	cadena Longitud máx.: 2048	El mensaje asociado con cualquier error encontrado al realizar esta comprobación durante esta auditoría.

Errores

InvalidRequestException

El contenido de la solicitud no es válido.

ResourceNotFoundException

El recurso especificado no existe.

ThrottlingException

El índice supera el límite.

InternalFailureException

Se ha producido un error inesperado.

ListAuditTasks

Muestra una lista de las auditorías de Device Defender que se han realizado durante un periodo de tiempo determinado.

Sinopsis

```
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 cli-input-json

```
{
  "startTime": "timestamp",
  "endTime": "timestamp",
  "taskType": "string",
  "taskStatus": "string",
  "nextToken": "string",
  "maxResults": "integer"
}
```

Campos cli-input-json

Name (Nombre)	Tipo	Descripción
startTime	timestamp	El comienzo del periodo de tiempo. La información de auditoría se conserva durante un tiempo limitado (180 días). Si se solicita una hora de inicio anterior al tiempo de retención, se produce la excepción <code>InvalidRequestException</code> .
endTime	timestamp	El final del periodo de tiempo.
taskType	string	Un filtro para limitar el resultado al tipo de auditoría especificado: puede ser uno entre <code>ON_DEMAND_AUDIT_TASK</code> o <code>SCHEDULED_AUDIT_TASK</code> . enum: <code>ON_DEMAND_AUDIT_TASK</code> <code>SCHEDULED_AUDIT_TASK</code>
taskStatus	string	Un filtro para limitar el resultado a las auditorías con el estado de finalización especificado: puede ser uno entre <code>IN_PROGRESS</code> , <code>COMPLETED</code> , <code>FAILED</code> o <code>CANCELED</code> . enum: <code>IN_PROGRESS</code> <code>COMPLETADO</code> <code>ERROR</code> <code>CANCELADO</code>
nextToken	string	El token del conjunto siguiente de resultados.
maxResults	entero Rango máx.: 250; mín.: 1	El número máximo de resultados que devolver a la vez. El valor predeterminado es 25.

Salida

```
{
  "tasks": [
```

```
{
    "taskId": "string",
    "taskStatus": "string",
    "taskType": "string"
}
],
"nextToken": "string"
}
```

Campos de salida de la CLI

Name (Nombre)	Tipo	Descripción
tareas	Lista miembro: AuditTaskMetadata Clase de Java: java.util.List	Las auditorías que se realizaron durante el periodo de tiempo especificado.
taskId	string Longitud máx.: 40; mín.: 1 Patrón: [a-zA-Z0-9-]+	El ID de esta auditoría.
taskStatus	string El estado de esta auditoría: uno entre IN_PROGRESS, COMPLETED, FAILED o CANCELED. enum: IN_PROGRESS COMPLETADO ERROR CANCELADO	
taskType	string El tipo de esta auditoría: uno entre ON_DEMAND_AUDIT_TASK o SCHEDULED_AUDIT_TASK. enum: ON_DEMAND_AUDIT_TASK SCHEDULED_AUDIT_TASK	
nextToken	string Un token que se puede utilizar para recuperar el siguiente conjunto de resultados o null si no hay resultados adicionales.	

Errores

InvalidRequestException

El contenido de la solicitud no es válido.

ThrottlingException

El índice supera el límite.

InternalFailureException

Se ha producido un error inesperado.

CancelAuditTask

Cancela una auditoría que está en curso. La auditoría puede ser programada o bajo demanda. Si la auditoría no está en curso, se produce la excepción `InvalidRequestException`.

Sinopsis

```
aws iot cancel-audit-task \
  --task-id <value> \
  [--cli-input-json <value>] \
  [--generate-cli-skeleton]
```

Formato `cli-input-json`

```
{  
    "taskId": "string"  
}
```

Campos `cli-input-json`

Name (Nombre)	Tipo	Descripción
taskId	string Longitud máx.: 40; mín.: 1 Patrón: [a-zA-Z0-9-]+	ID de la auditoría que desea cancelar. Solo puede cancelar una auditoría que esté IN_PROGRESS.

Salida

Ninguno

Errores

`ResourceNotFoundException`

El recurso especificado no existe.

`InvalidRequestException`

El contenido de la solicitud no es válido.

`ThrottlingException`

El índice supera el límite.

`InternalFailureException`

Se ha producido un error inesperado.

Comprobar resultados de auditoría

Utilice `ListAuditFindings` para ver los resultados de una auditoría. Puede filtrar los resultados por el tipo de comprobación, por recurso específico o por la hora de la auditoría. Puede utilizar esta información para mitigar cualquier problema que se encuentre.

Puede definir acciones de mitigación y aplicarlas a los resultados de su auditoría. Para obtener más información, consulte [Acciones de mitigación \(p. 1036\)](#).

ListAuditFindings

Muestra una lista de los hallazgos (resultados) de una auditoría de Device Defender o de las auditorías realizadas durante un periodo de tiempo especificado. (Los hallazgos se conservan durante 180 días).

Sinopsis

```
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 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 (Nombre)	Tipo	Descripción
taskId	string	Un filtro para limitar los resultados a la auditoría que tiene el ID especificado. Debe especificar taskId o startTime y endTime, pero no ambos. Longitud máx.: 40; mín.: 1 Patrón: [a-zA-Z0-9-]+
checkName	string	Un filtro para limitar los resultados a los hallazgos de la verificación de auditoría especificada.
resourceIdentifier	ResourceIdentifier	La información que identifica el recurso no conforme.

Name (Nombre)	Tipo	Descripción
deviceCertificateId	string Longitud máx.: 64; mín.: 64 Patrón: (0x)?[a-fA-F0-9]+	El ID del certificado asociado al recurso.
caCertificateId	string Longitud máx.: 64; mín.: 64 Patrón: (0x)?[a-fA-F0-9]+	El ID del certificado de CA utilizado para autorizar el certificado.
cognitoIdentityPoolId	string	El ID del grupo de identidades de Amazon Cognito.
clientId	string	El ID de cliente.
policyVersionIdentifier	PolicyVersionIdentifier	La versión de la política asociada a este recurso.
policyName	string Longitud máx.: 128; mín.: 1 patrón: [w+=,.@-]+	El nombre de la política.
policyVersionId	string Pattern: [0-9] +	El ID de la versión de la política asociada a este recurso.
roleAliasArn	string	El ARN del alias de rol que tiene acciones excesivamente permisivas. Longitud: máx.: 2048; mín.: 1
account	cadena Longitud máx.: 12; mín.: 12 Pattern: [0-9] +	La cuenta a la que está asociado el recurso.
maxResults	entero Rango máx.: 250; mín.: 1	El número máximo de resultados que devolver a la vez. El valor predeterminado es 25.
nextToken	string	El token del conjunto siguiente de resultados.
startTime	timestamp	Un filtro para limitar los resultados a los encontrados después de la hora especificada. Debe especificar startTime y endTime o taskId, pero no ambos.

Name (Nombre)	Tipo	Descripción
endTime	timestamp	Un filtro para limitar los resultados a los encontrados antes de la hora especificada. Debe especificar startTime y endTime o taskId, pero no ambos.

Salida

```
{
  "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 salida de la CLI

Name (Nombre)	Tipo	Descripción
findings	Lista miembro: AuditFinding	Los hallazgos (resultados) de la auditoría.
taskId	string Longitud máx.: 40; mín.: 1 Patrón: [a-zA-Z0-9-]+	El ID de la auditoría que generó este resultado (hallazgo).
checkName	string	La comprobación de auditoría que generó este resultado.
taskStartTime	timestamp	La hora a la que se inició la auditoría.
findingTime	timestamp	La hora a la que se descubrió el resultado (hallazgo).
severity	string enum: CRÍTICO ALTO MEDIO BAJO	La gravedad del resultado (hallazgo).
nonCompliantResource	NonCompliantResource	El recurso que se ha comprobado que no cumple los requisitos de una comprobación de auditoría.
resourceType	string enum: DEVICE_CERTIFICATE CA_CERTIFICATE IOT_POLICY COGNITO_IDENTITY_POOL CLIENT_ID ACCOUNT_SETTINGS	El tipo del recurso no conforme. enum: DEVICE_CERTIFICATE CA_CERTIFICATE IOT_POLICY COGNITO_IDENTITY_POOL CLIENT_ID ACCOUNT_SETTINGS
resourceIdentifier	ResourceIdentifier	La información que identifica el recurso no conforme.
deviceCertificateId	string Longitud máx.: 64; mín.: 64 Patrón: (0x)?[a-fA-F0-9]+	El ID del certificado asociado al recurso.
caCertificateId	string Longitud máx.: 64; mín.: 64	El ID del certificado de CA utilizado para autorizar el certificado.

Name (Nombre)	Tipo	Descripción
	Patrón: (0x)?[a-fA-F0-9]+	
cognitoidentityPoold	string	El ID del grupo de identidades de Amazon Cognito.
clientId	string	El ID de cliente.
policyVersionIdentifier	PolicyVersionIdentifier	La versión de la política asociada a este recurso.
policyName	string Longitud máx.: 128; mín.: 1 patrón: [w+=,.@-]+	El nombre de la política.
policyVersionId	string Pattern: [0-9] +	El ID de la versión de la política asociada a este recurso.
cuenta	cadena Longitud máx.: 12; mín.: 12 Pattern: [0-9] +	La cuenta a la que está asociado el recurso.
additionalInfo	map	Otra información sobre el recurso no conforme.
relatedResources	Lista miembro: RelatedResource	La lista de recursos relacionados.
resourceType	string enum: DEVICE_CERTIFICATE CA_CERTIFICATE IOT_POLICY COGNITO_IDENTITY_POOL CLIENT_ID ACCOUNT_SETTINGS	El tipo de recurso.
resourceIdentifier	ResourceIdentifier	Información que identifica el recurso.
deviceCertificateId	string Longitud máx.: 64; mín.: 64 Patrón: (0x)?[a-fA-F0-9]+	El ID del certificado asociado al recurso.
caCertificateId	string Longitud máx.: 64; mín.: 64 Patrón: (0x)?[a-fA-F0-9]+	El ID del certificado de CA utilizado para autorizar el certificado.
cognitoidentityPoold	string	El ID del grupo de identidades de Amazon Cognito.

Name (Nombre)	Tipo	Descripción
clientId	string	El ID de cliente.
policyVersionIdentifier	PolicyVersionIdentifier	La versión de la política asociada a este recurso.
iamRoleArn	string Longitud máx.: 2048; mín.: 20	El ARN del rol de IAM que tiene acciones excesivamente permisivas.
policyName	string Longitud máx.: 128; mín.: 1 patrón: [w+=,.@-]+	El nombre de la política.
policyVersionId	string Pattern: [0-9] +	El ID de la versión de la política asociada a este recurso.
roleAliasArn	string Longitud: máx.: 2048; mín.: 1	El ARN del alias de rol que tiene acciones excesivamente permisivas.
cuenta	cadena Longitud máx.: 12; mín.: 12 Pattern: [0-9] +	La cuenta a la que está asociado el recurso.
additionalInfo	map	Otra información sobre el recurso.
reasonForNonCompliance	string	El motivo por el que el recurso no era conforme.
reasonForNonComplianceCode	string	Un código que indica el motivo por el cual el recurso no era conforme.
nextToken	string	Un token que se puede utilizar para recuperar el siguiente conjunto de resultados o null si no hay resultados adicionales.

Errores

InvalidRequestException

El contenido de la solicitud no es válido.

ThrottlingException

El índice supera el límite.

InternalFailureException

Se ha producido un error inesperado.

Supresiones de hallazgos de auditoría

Cuando ejecuta una auditoría, informa de los hallazgos de todos los recursos no conformes. Esto significa que los informes de auditoría incluyen los resultados de los recursos en los que está trabajando para mitigar problemas y también para recursos que se sabe que no cumplen los requisitos, como dispositivos de prueba o rotos. La auditoría continúa informando de los resultados de los recursos que siguen sin cumplir en las sucesivas ejecuciones de auditoría, lo que puede agregar información no deseada a los informes. Las supresiones de búsquedas de auditoría permiten suprimir o filtrar los hallazgos durante un período de tiempo definido hasta que se corrija el recurso o indefinidamente para un recurso asociado a una prueba o dispositivo roto.

Note

Las acciones de mitigación no estarán disponibles para los hallazgos de auditoría suprimidos. Para obtener más información sobre las acciones de mitigación, consulte [Acciones de mitigación \(p. 1036\)](#).

Para obtener información sobre cuotas de supresión de hallazgos de auditoría, consulte [AWS IoT Cuotas y puntos de enlace de Device Defender](#).

Cómo funcionan las supresiones de búsqueda de auditorías

Cuando crea una supresión de búsqueda de auditoría para un recurso que no cumple los requisitos, los informes de auditoría y las notificaciones se comportan de forma diferente.

Los informes de auditoría incluirán una nueva sección que enumera todos los hallazgos suprimidos asociados al informe. Los hallazgos suprimidos no se tendrán en cuenta cuando evaluamos si una comprobación de auditoría cumple las normas o no. También se devuelve un recuento de recursos suprimidos para cada comprobación de auditoría cuando se utiliza el `describe-audit-task` en la interfaz de línea de comandos (CLI).

Para las notificaciones de auditoría, los hallazgos suprimidos no se tienen en cuenta cuando evaluamos si una comprobación de auditoría cumple las normas o no. También se incluye un recuento de recursos suprimido en cada notificación de comprobación de auditoría AWS IoT Device Defender pública en Amazon CloudWatch y Amazon Simple Notification Service (Amazon SNS).

Cómo utilizar las supresiones de búsqueda de auditoría en la consola

Para suprimir un hallazgo de un informe de auditoría

El siguiente procedimiento muestra cómo crear una supresión de hallazgos de auditoría en AWS IoT consola de .

1. En el navegador [AWS IoT consola](#), en el panel de navegación, expanda `Defender` y luego seleccione `Auditoría`, `Resultados`.
2. Selecciona un informe de auditoría que quieras revisar.

The screenshot shows the AWS IoT Device Defender Audit Results interface. On the left is a navigation sidebar with options like Monitor, Activity, Onboard, Manage, Greengrass, Secure, Defend, Audit (which is expanded to show Results, Schedules, Action executions, Finding suppressions, Detect, Mitigation actions, and Settings), and Detect. The main area is titled "Audit results (10+)" and contains a table with columns: Name, Date, Status, and Summary. The table lists 14 audit runs, all of which are marked as "Not compliant" (indicated by a red triangle icon). The "Status" column shows a green circle for one run and a red triangle for the others. The "Summary" column indicates "1 of 14 non-compliant" for each row.

Name	Date	Status	Summary
On-demand	July 28, 2020, 14:14:18 (UTC-0700)	⚠️ Not compliant	1 of 14 non-compliant
On-demand	July 28, 2020, 11:55:43 (UTC-0700)	🟢 Compliant	14 of 14 completed
AWSIoTDeviceDefenderDailyAudit	July 28, 2020, 05:12:39 (UTC-0700)	⚠️ Not compliant	1 of 14 non-compliant
AWSIoTDeviceDefenderDailyAudit	July 27, 2020, 05:12:39 (UTC-0700)	⚠️ Not compliant	1 of 14 non-compliant
AWSIoTDeviceDefenderDailyAudit	July 26, 2020, 05:12:39 (UTC-0700)	⚠️ Not compliant	1 of 14 non-compliant
AWSIoTDeviceDefenderDailyAudit	July 25, 2020, 05:12:39 (UTC-0700)	⚠️ Not compliant	1 of 14 non-compliant
AWSIoTDeviceDefenderDailyAudit	July 24, 2020, 05:12:39 (UTC-0700)	⚠️ Not compliant	1 of 14 non-compliant
AWSIoTDeviceDefenderDailyAudit	July 23, 2020, 05:12:39 (UTC-0700)	⚠️ Not compliant	1 of 14 non-compliant
AWSIoTDeviceDefenderDailyAudit	July 22, 2020, 05:12:39 (UTC-0700)	⚠️ Not compliant	1 of 14 non-compliant
AWSIoTDeviceDefenderDailyAudit	July 21, 2020, 05:12:39 (UTC-0700)	⚠️ Not compliant	1 of 14 non-compliant

3. En el navegadorComprobaciones no conformessección, enNombre de la verificación, elija la comprobación de auditoría que le interese.

The screenshot shows the AWS IoT Device Defender Audit Report interface. At the top, there is a breadcrumb navigation: AWS IoT > Device Defender > Audit > Audit Results > Audit Report. Below the navigation, the title "Audit Report" is displayed, followed by the timestamp "On-demand - July 28, 2020, 14:14:18 (UTC-0700)".

Audit findings:

- Audit task ID: 40c1204d7be8bb0d33682ef35c144231
- Started at: July 28, 2020, 14:14:18 (UTC-0700)

Non-compliant checks (1 of 14):

Check name	Severity	Non-compliant resources	% Resources	Mitigation
Logging disabled	Low	1	100%	Logging disabled ⓘ

Compliant checks (13 of 14):

Check name	Severity	Scanned ⓘ
Authenticated Cognito role overly permissive	Critical	0
CA certificate key quality	Critical	0
CA certificate revoked but device certificates still active	Critical	0
Device certificate key quality	Critical	0
Device certificate shared	Critical	0
IoT policies overly permissive	Critical	0
Role alias overly permissive	Critical	0
Unauthenticated Cognito role overly permissive	Critical	0
Conflicting MQTT client IDs	High	0
CA certificate expiring	Medium	0
Device certificate expiring	Medium	0
Revoked device certificate still active	Medium	0
Role alias allows access to unused services	Medium	0

4. En la pantalla de detalles de la comprobación de auditoría, si hay hallazgos que no quieras ver, selecciona el botón de opción junto al hallazgo. A continuación, elija Actions, a continuación, elija la cantidad de tiempo que desea que persista la supresión de búsqueda de auditorías.

Note

En la consola de puede seleccionar 1 semana, 1 mes, 3 meses, 6 meses, o bien indefinidamente como fechas de caducidad para la supresión de búsqueda de auditorías. Si quieres establecer una fecha de caducidad específica, solo puedes hacerlo en la CLI o la API. Las supresiones de búsqueda de auditorías también se pueden cancelar en cualquier momento, independientemente de la fecha de caducidad.

The screenshot shows the AWS IoT Device Defender Audit Findings page. The navigation path is: AWS IoT > Device Defender > Audit > Audit Results > Audit Report > Audit Findings. The title is "Audit Findings" and it says "Logging disabled". A box indicates "1 account non-compliant". Under "Mitigation", it says "Enable CloudWatch Logs.". Below this, a table lists "Non-compliant account (1)". The table has columns: Finding, Reason, and Account settings. One row is shown: "417b2f816eac7a2e40fdb0bc709b01a2" with reason "Logging disabled on account." and account ID "765219403047". To the right of the table is a "Actions" dropdown menu with options: Start mitigation actions, Suppress Finding, 1 week, 1 month, 3 months, 6 months, and Indefinitely.

5. Confirme los detalles de supresión y luego seleccioneHabilitar supresión.

The screenshot shows a "Confirm suppression" dialog box. It asks "Please verify the details of the audit finding suppression". The details listed are: Check name "Logging disabled", Account settings "765219403047", Expiration period "3 months", and Expiration date "2020-10-28T21:25:41.100Z". At the bottom are "Cancel" and "Enable suppression" buttons, with "Enable suppression" being orange.

6. Después de crear la supresión de búsqueda de auditoría, aparece un banner que confirma que se ha creado la supresión de búsqueda de auditoría.

The screenshot shows the AWS IoT Audit Findings interface. At the top, a green banner displays the message: "Audit finding suppression created successfully" and "The finding related to the resource is suppressed for audit check Logging disabled". Below the banner, the navigation path is: AWS IoT > Device Defender > Audit > Audit Results > Audit Report > Audit Findings. The main title is "Audit Findings" with a subtitle "Logging disabled". A section titled "1 account non-compliant" contains a mitigation note: "Enable CloudWatch Logs.". Below this is a table titled "Non-compliant account (1)". The table has columns: "Finding", "Reason", and "Account settings". One row is shown: "417b2f816eac7a2e40fdb0bc709b01a2" (Finding), "Logging disabled on account." (Reason), and "765219403047" (Account settings). There are "Actions" and navigation buttons (< 1 >) at the top right of the table.

Para ver los hallazgos suprimidos en un informe de auditoría

1. En el navegador [AWS IoT consola](#), en el panel de navegación, expanda **Defender** y luego seleccione **Auditoría**, **Resultados**.
2. Selecciona un informe de auditoría que quieras revisar.
3. En el navegador **Conclusiones suprimidas**, ver qué resultados de auditoría se han suprimido para el informe de auditoría elegido.

The screenshot shows the AWS IoT Device Defender Audit Report interface. The left sidebar navigation includes sections like Monitor, Activity, Onboard, Manage, Greengrass, Secure, Defend (with Intro selected), Audit (with Results, Schedules, Action executions, Finding suppressions), Detect (with Mitigation actions), and Act (with Test). The main content area displays an Audit Report for an on-demand audit on July 28, 2020, at 11:55:43 UTC. It lists 14 compliant checks (including Authenticated Cognito role overly permissive, CA certificate key quality, etc.) and 1 suppressed finding (Logging disabled). A search bar at the bottom allows filtering by check name.

Check name	Severity	Scanned
Authenticated Cognito role overly permissive	Critical	0
CA certificate key quality	Critical	0
CA certificate revoked but device certificates still active	Critical	0
Device certificate key quality	Critical	0
Device certificate shared	Critical	0
IoT policies overly permissive	Critical	0
Role alias overly permissive	Critical	0
Unauthenticated Cognito role overly permissive	Critical	0
Conflicting MQTT client IDs	High	0
CA certificate expiring	Medium	0
Device certificate expiring	Medium	0
Revoked device certificate still active	Medium	0
Role alias allows access to unused services	Medium	0
Logging disabled	Low	1

Suppressed findings (1)			
<input type="text"/> Filter suppressions by check name			
Check name	Finding	Reason	Resource identifier
Logging disabled	755a27914fb2ca24a8b3d47ef3563726	Logging disabled on account.	765219403047

Para enumerar las supresiones de búsquedas de auditorías

- En el navegador [AWS IoT consola](#), en el panel de navegación, expanda **Defender** y luego seleccione **Auditoría**. Encontrar supresión.

The screenshot shows the AWS IoT Device Defender Audit Finding Suppressions interface. On the left is a navigation sidebar with sections like Monitor, Activity, Onboard, Manage, Greengrass, Secure, Defend, Audit (which is expanded to show Results, Schedules, Action executions, and Finding suppressions), Detect, Mitigation actions (new), Settings, Act, and Test. The main content area has a breadcrumb trail: AWS IoT > Device Defender > Audit > Audit Finding Suppressions. It displays a table titled "Audit finding suppressions (1) Info". The table has columns: Resource identifier, Check name, Expiration date, and Description. A single row is shown: Resource identifier is 765219403047, Check name is Logging disabled, Expiration date is October 28, 2020, 14:26:53 (UTC-0700), and Description is -. There are "Actions" and "Create" buttons at the top right of the table.

Para editar la supresión de búsqueda de auditorías

1. En el navegador [AWS IoT consola](#), en el panel de navegación, expanda **Defend** y luego seleccione **Auditoría**, **Encontrar supresión**.
2. Seleccione el botón de opción situado junto a la supresión de hallazgos de auditoría que desea editar. A continuación, elija **Acciones**, **Editar**.
3. En la página **Editar supresión de búsqueda de auditoría** ventana, puede cambiar la **Duración de supresión** o **Descripción** (opcional).

Edit audit finding suppression

Suppressing an audit finding on a specified resource means that the finding related to the resource for the specified audit check will no longer be flagged as non-compliant.

Audit check

Logging disabled

Resource identifier

Account ID
765219403047

Suppression duration

The expiration date is October 28, 2020, 14:26:53 (UTC-0700). Select a different duration to change this.

6 months

Description (optional)

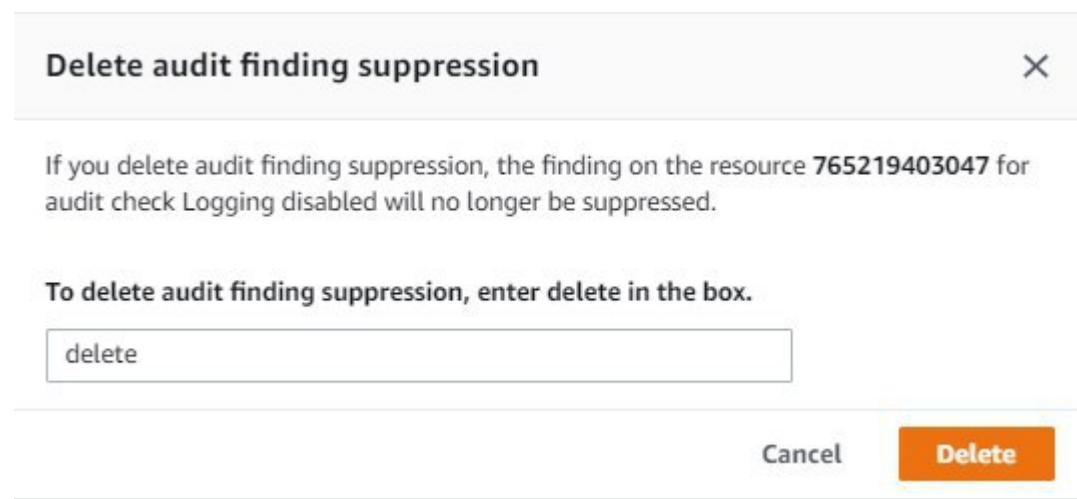
Suppresses "Logging disabled" check because I don't want to enable logging for now.

Cancel **Save**

4. Después de hacer los cambios, elija Guardar. La Encontrar supresión Se abrirá una ventana.

Para eliminar una supresión de búsqueda de auditoría

1. En el navegador [AWS IoT consola](#), en el panel de navegación, expanda Defendery luego seleccione Auditoría, Encontrar supresión.
2. Seleccione el botón de opción situado junto a la supresión de hallazgos de auditoría que desea eliminar Actions, Borrar.
3. En la página Eliminar supresión de búsqueda de auditoría ventana, introduzca el texto para confirmar la eliminación y, a continuación, elija Borrar. La Encontrar supresión Se abrirá una ventana.



Cómo utilizar las supresiones de búsqueda de auditoría en la CLI

Puede utilizar los siguientes comandos de la CLI de la para crear y administrar supresión de hallazgos de auditoría.

- [create-audit-supression](#)
- [describe-audit-supression](#)
- [actualización y supresión de auditorías](#)
- [supresión de eliminación de auditorías](#)
- [lista-auditorías-supresiones](#)

La `resource-identifier` entrada depende de la `check-name` para lo que estás suprimiendo los hallazgos. En la tabla siguiente se detallan las comprobaciones que requieren `resource-identifier` para crear y editar supresiones.

Note

Los comandos de supresión no indican desactivar una auditoría. Las auditorías seguirán ejecutándose en tu AWS IoT dispositivos. Las supresiones solo se aplican a las conclusiones de la auditoría.

check-name	resource-identifier
AUTHENTICATE_COGNITO_ROLE_OVERLY_PERMISSIVE_CHECK	deviceIdentityPoolId
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

check-name	resource-identifier
IOT_POLICY_OVERLY_PERMISSIVE_CHECK	policyVersionIdentifier
IOT_ROLE_ALIAS_ALLows_ACCESS_TO_UNUSED_SERVICES_CHECK	roleAliasArn
LOGGING_DISABLED_CHECK	account
REVOKEd_CA_CERT_CHECK	caCertificateId
REVOKEd_DEVICE_CERT_CHECK	deviceCertificateId
UNAUTHENTICATED_COGNITO_ROLE_OVERLY_PERMISSIVE_CHECK	identityPoolId

Para crear y aplicar una supresión de búsqueda de auditoría

El siguiente procedimiento muestra cómo crear una supresión de hallazgos de auditoría en AWS CLI.

- Usar `create-audit-suppression` para crear una supresión de búsqueda de auditoría. En el ejemplo siguiente se crea una supresión de búsqueda de auditoría para Cuenta de AWS [123456789012](#) sobre la base del cheque Registro deshabilitado.

```
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."
```

No se obtienen resultados de este comando.

API de supresiones de búsqueda de auditoría

Las siguientes API se pueden utilizar para crear y administrar supresiones de búsqueda de auditorías.

- [Crear supresión de auditoría](#)
- [Describir la supresión de auditorías](#)
- [Actualización de supresión de auditoría](#)
- [Eliminar supresión de auditoría](#)
- [Listar supresiones de auditoría](#)

Para filtrar para hallazgos de auditoría específicos, puede utilizar el [ListAuditFindings API](#).

Detect

AWS IoT Device Defender Detect permite identificar comportamientos inusuales que podrían indicar que un dispositivo se ha visto comprometido cuando se monitorea el comportamiento de los dispositivos. Si utiliza una combinación de métricas del lado de la nube (procedentes de AWS IoT) y de métricas del lado del dispositivo (procedentes de agentes instalados en los dispositivos), puede detectar:

- Cambios en los patrones de conexión.
- Dispositivos que se comunican con puntos de enlace no autorizados o no reconocidos.
- Cambios en los patrones de tráfico de entrada y salida de los dispositivos.

Cree perfiles de seguridad, que contengan definiciones de comportamientos esperados de los dispositivos y asígnelos a un grupo de dispositivos o a todos los dispositivos de su flota. AWS IoT Device Defender Detect utiliza estos perfiles de seguridad para detectar anomalías y enviar alarmas a través de métricas de Amazon CloudWatch y notificaciones de Amazon Simple Notification Service.

AWS IoT Device Defender Detect puede detectar problemas de seguridad que se producen con frecuencia en los dispositivos conectados:

- Tráfico desde un dispositivo a una dirección IP maliciosa conocida o a un punto de enlace no autorizado que indica un posible comando y canal de control maliciosos.
- Tráfico anómalo, como un pico en el tráfico saliente, que indica que un dispositivo participa en un DDoS.
- Dispositivos con puertos e interfaces de administración remota a los que se puede acceder de forma remota.
- Un pico en el índice de mensajes que se envían a la cuenta (por ejemplo, desde un dispositivo fraudulento, lo que podría producir unos gastos excesivos por mensaje).

Casos de uso

Medir la superficie de ataque

Puede usar AWS IoT Device Defender Detect para medir la superficie de ataque de sus dispositivos. Puede, por ejemplo, identificar dispositivos con puertos de servicio que con frecuencia son objeto de campañas de ataques (el servicio telnet que se ejecuta en los puertos 23/2323, el servicio SSH que se ejecuta en el puerto 22, los servicios HTTP/S que se ejecutan en los puertos 80/443/8080/8081). Aunque estos puertos de servicio pueden tener motivos legítimos para utilizarse en los dispositivos, también suelen formar parte de la superficie de ataque de los adversarios y llevan asociados riesgos. Después AWS IoT Device Defender Detecte alarmas de la superficie de ataque, esta puede minimizarse (eliminando servicios de red que no se utilizan) o se pueden ejecutar otras evaluaciones para detectar debilidades de seguridad (por ejemplo, telnet configurado con contraseñas comunes, predeterminadas o poco seguras).

Detectar anomalías de comportamiento de dispositivos con posibles causas raíces de seguridad

Puede usar AWS IoT Device Defender Detect para alertarle sobre las métricas de comportamiento inesperado del dispositivo (el número de puertos abiertos, el número de conexiones, un puerto abierto inesperado, las conexiones a direcciones IP inesperadas) que pueden indicar una vulneración de la seguridad. Por ejemplo, un número de conexiones TCP más alto de lo esperado puede indicar que un dispositivo se está utilizando para un ataque DDoS. Un proceso que se escucha en un puerto diferente al que espera puede indicar una puerta trasera instalada en un dispositivo para control remoto. Puede usar AWS IoT Device Defender Detect para sondear el estado de las flotas de dispositivos y contrastar los supuestos de seguridad (por ejemplo, que ningún dispositivo está escuchando en el puerto 23 o 2323).

Puede habilitar la detección de amenazas basada en aprendizaje automático (ML) para identificar automáticamente posibles amenazas.

Detectar un dispositivo configurado incorrectamente

Un pico en el número o tamaño de los mensajes enviados desde un dispositivo a su cuenta puede indicar un dispositivo configurado incorrectamente. Un dispositivo como este podría aumentar los gastos por mensaje. Del mismo modo, un dispositivo con muchos errores de autorización podría requerir la reconfiguración de una política.

Monitorización del comportamiento de dispositivos no registrados

AWS IoT Device Defender Detect permite identificar comportamientos inusuales en dispositivos que no figuran en el registro de AWS IoT. Puede definir perfiles de seguridad que sean específicos de uno de los siguientes tipos de destino:

- Todos los dispositivos
- Todos los dispositivos registrados (objetos en el registro de AWS IoT)
- Todos los dispositivos no registrados
- Los dispositivos de un grupo de objetos

Un perfil de seguridad define un conjunto de comportamientos esperados para los dispositivos de su cuenta y especifica las acciones que se realizarán cuando se detecte una anomalía. Los perfiles de seguridad deben asociarse al mayor número de destinos específicos para controlar con detalle qué dispositivos se están evaluando en ese perfil.

Los dispositivos no registrados deben proporcionar un identificador de cliente de MQTT o un nombre de objeto coherentes (para los dispositivos que registran métricas de dispositivo) durante todo el ciclo de vida del dispositivo, de forma que todas las infracciones y métricas se atribuyan al mismo dispositivo.

Important

Los mensajes registrados por los dispositivos se rechazan si el nombre del objeto contiene caracteres de control o si el nombre del objeto tiene más de 128 bytes de caracteres con codificación UTF-8.

Casos de uso de seguridad

En esta sección se describen los distintos tipos de ataques que amenazan la flota de dispositivos y las métricas recomendadas que puede utilizar para supervisar estos ataques. Recomendamos utilizar anomalías métricas como punto de partida para investigar los problemas de seguridad, pero no debe basar la determinación de ninguna amenaza de seguridad únicamente en una anomalía métrica.

Para investigar una alarma de anomalía, correlacione los detalles de la alarma con otra información contextual, como atributos de dispositivo, tendencias históricas de métricas de dispositivos, tendencias históricas de métricas de perfil de seguridad, métricas personalizadas y registros para determinar si existe una amenaza a la seguridad.

Casos de uso del lado de la nube

Device Defender puede supervisar los siguientes casos de uso en el AWS IoT lado nube.

Robo de propiedad intelectual:

El robo de propiedad intelectual implica robar las propiedades intelectuales de una persona o empresa, incluidos secretos comerciales, hardware o software. A menudo ocurre durante la etapa de fabricación de los dispositivos. El robo de propiedad intelectual puede producirse en forma de piratería, robo de dispositivos o robo de certificados de dispositivos. El robo de propiedad intelectual basado en la nube puede producirse debido a la presencia de políticas que permiten el acceso no deseado a los recursos de IoT. Debería revisar la aplicación [Políticas de IoT dey encienda Auditar comprobaciones excesivamente permisivas](#) para identificar políticas excesivamente permisivas.

Métricas relacionadas:

Métrica	Justificación
IP de origen	Si se roba un dispositivo, su dirección IP de origen quedaría fuera del rango de direcciones IP normalmente esperado para los dispositivos que circulan en una cadena de suministro normal.
Número de mensajes recibidos	Dado que un atacante puede utilizar un dispositivo en caso de robo de IP basado en la nube, métricas relacionadas con recuentos de mensajes o tamaños de mensajes enviados al dispositivo desde AWS IoT a la nube pueden aumentar, lo que indica un posible problema de seguridad.
Tamaño del mensaje	

Exfiltración de datos basada en MQTT:

La exfiltración de datos se produce cuando un actor malintencionado realiza una transferencia de datos no autorizada desde una implementación de IoT o desde un dispositivo. El atacante lanza este tipo de ataques a través de MQTT contra fuentes de datos del lado de la nube.

Métricas relacionadas:

Métrica	Justificación
IP de origen	Si un dispositivo es robado, su dirección IP de origen quedaría fuera del rango de direcciones IP normalmente esperado para los dispositivos que circulan en una cadena de suministro estándar.
Número de mensajes recibidos	Dado que un atacante puede utilizar un dispositivo en una exfiltración de datos basada en MQTT, métricas relacionadas con recuentos de mensajes o tamaños de mensajes enviados al dispositivo desde AWS IoT a la nube pueden aumentar, lo que indica un posible problema de seguridad.
Tamaño del mensaje	

Suplantación

Un ataque de suplantación es cuando los atacantes se hacen pasar por entidades conocidas o de confianza en un esfuerzo por acceder a servicios, aplicaciones, datos o participar en el mando y el control de dispositivos IoT.

Métricas relacionadas:

Métrica	Justificación
Errores de autorización	Cuando los atacantes se hacen pasar por entidades de confianza mediante el uso de identidades robadas, las métricas relacionadas con la conectividad suelen aumentar, ya
Intentos de conexión	
Desconecta	

Métrica	Justificación
	que es posible que las credenciales ya no sean válidas o ya las utilice un dispositivo de confianza. Los comportamientos anómalos en fallos de autorización, intentos de conexión o desconexiones apuntan a un escenario de suplantación potencial.

Abuso de infraestructura cloud:

Abuso a AWS IoT los servicios en la nube se producen al publicar o suscribirse a temas con un gran volumen de mensajes o con mensajes de gran tamaño. Las políticas excesivamente permisivas o el aprovechamiento de vulnerabilidades de dispositivos para el mando y el control también pueden provocar un abuso de la infraestructura en la nube. Uno de los principales objetivos de este ataque es aumentar tu AWSfactura. Debería revisar la aplicación [Políticas de IoT de y encienda Auditar comprobaciones excesivamente permisivas](#) para identificar políticas excesivamente permisivas.

Métricas relacionadas:

Métrica	Justificación
Número de mensajes recibidos	El objetivo de este ataque es aumentar tu AWSfactura, las métricas que supervisan actividades como el recuento de mensajes, los mensajes recibidos y el tamaño de los mensajes aumentarán.
Número de mensajes enviados	
Tamaño del mensaje	
IP de origen	Pueden aparecer listas de IP de origen sospechosas, a partir de las cuales los atacantes generan su volumen de mensajería.

Casos de uso del lado del dispositivo

Device Defender puede supervisar los siguientes casos de uso en el lado del dispositivo.

Ataque de denegación de servicio:

Un ataque de denegación de servicio (DoS) tiene como objetivo apagar un dispositivo o una red, lo que hace que el dispositivo o la red sean inaccesibles para los usuarios previstos. Los ataques DoS bloquean el acceso inundando el destino de tráfico o enviándole solicitudes que inicien la ralentización de un sistema o provocan que el sistema falle. Los dispositivos IoT se pueden utilizar en ataques DoS.

Métricas relacionadas:

Métrica	Justificación
Paquetes fuera	Los ataques DoS suelen implicar mayores velocidades de comunicación saliente desde un dispositivo determinado y, según el tipo de ataque DoS, podría haber un aumento en uno o ambos del número de paquetes salientes y bytes de salida.
Bytes de salida	
IP de destino	Si define las direcciones IP/rangos CIDR con los que deben comunicarse los dispositivos,

Métrica	Justificación
	una anomalía en la IP de destino puede indicar una comunicación IP no autorizada desde sus dispositivos.
Puertos TCP de escucha	
Recuento de puertos TCP de escucha	
Puertos UDP de escucha	
Recuento de puertos UDP de escucha	Un ataque DoS suele requerir una infraestructura de mando y control más grande en la que el malware instalado en los dispositivos recibe comandos e información sobre quién atacar y cuándo atacar. Por lo tanto, para recibir dicha información, el malware suele escuchar en puertos que los dispositivos no utilizan normalmente.

Intensificación de amenazas laterales:

La escalada de amenazas laterales suele comenzar cuando un atacante obtiene acceso a un punto de una red, por ejemplo, a un dispositivo conectado. El atacante intenta aumentar su nivel de privilegios o su acceso a otros dispositivos mediante métodos tales como credenciales robadas o vulnerabilidades.

Métricas relacionadas:

Métrica	Justificación
Paquetes fuera	
Bytes de salida	En situaciones típicas, el atacante tendría que realizar un escaneo en la red de área local para realizar un reconocimiento e identificar los dispositivos disponibles para reducir su selección de objetivos de ataque. Este tipo de análisis podría dar lugar a un pico de bytes y recuentos de salidas de paquetes.
IP de destino	Si se supone que un dispositivo debe comunicarse con un conjunto conocido de direcciones IP o CIDR, puede identificar si intenta comunicarse con una dirección IP anormal, que a menudo sería una dirección IP privada en la red local en un caso de uso de escalada de amenazas laterales.
Errores de autorización	A medida que el atacante intenta aumentar su nivel de privilegios en una red de IoT, puede utilizar credenciales robadas que se han revocado o han caducado, lo que podría provocar mayores errores de autorización.

Exfiltración o vigilancia de datos:

La exfiltración de datos se produce cuando el malware o un actor malintencionado realizan una transferencia de datos no autorizada desde un dispositivo o un endpoint de red. La exfiltración de datos normalmente tiene dos fines para el atacante, obtener datos o propiedad intelectual, o realizar el reconocimiento de una red. Vigilancia significa que el código malicioso se utiliza para supervisar las actividades de los usuarios con el fin de robar credenciales y recopilar información. Las siguientes métricas pueden proporcionar un punto de partida para investigar cualquier tipo de ataque.

Métricas relacionadas:

Métrica	Justificación
Paquetes fuera	Cuando se producen ataques de exfiltración de datos o vigilancia, el atacante a menudo reflejaba los datos que se envían desde el dispositivo en lugar de simplemente redirigir los datos, lo que el defensor identificaría cuando no vean venir los datos previstos. Estos datos reflejados aumentarían significativamente la cantidad total de datos enviados desde el dispositivo, lo que provocaría un aumento de los recuentos de paquetes y bytes de salida.
Bytes de salida	
IP de destino	Cuando un atacante utiliza un dispositivo en ataques de exfiltración de datos o vigilancia, los datos tendrían que enviarse a una dirección IP anormal controlada por el atacante. La supervisión de la IP de destino puede ayudar a identificar dicho ataque.

Minería criptoCurrency

Los atacantes aprovechan la potencia de procesamiento de los dispositivos para extraer criptomonedas. La criptominería es un proceso de uso intensivo computacional, que normalmente requiere comunicación de red con otros pares y grupos de minería.

Métricas relacionadas:

Métrica	Justificación
IP de destino	La comunicación de red suele ser un requisito durante la criptominería. Tener una lista estrictamente controlada de direcciones IP con las que el dispositivo debe comunicarse puede ayudar a identificar la comunicación no deseada en un dispositivo, como la minería de criptomonedas.
CPU usage Métricas personalizadas de	La minería de criptomonedas requiere un cálculo intensivo que da como resultado una alta utilización de la CPU del dispositivo. Si elige recopilar y supervisar esta métrica, un uso de CPU superior a lo normal podría ser un indicador de actividades de criptominería.

Comando y control, malware y ransomware

El malware o el ransomware restringen el control sobre los dispositivos y limitan la funcionalidad de su dispositivo. En el caso de un ataque de ransomware, se perdería el acceso a los datos debido al cifrado que utiliza el ransomware.

Métricas relacionadas:

Métrica	Justificación
IP de destino	Los ataques remotos o de red representan una gran parte de los ataques contra dispositivos IoT. Una lista controlada de direcciones IP con las que el dispositivo debe comunicarse puede ayudar a identificar IP de destino anormales resultantes de un ataque de malware o ransomware.
Puertos TCP de escucha	Varios ataques de malware implican iniciar un servidor de comandos y control que envía comandos para ejecutarse en un dispositivo.
Recuento de puertos TCP de escucha	Este tipo de servidor es fundamental para una operación de malware o ransomware y se puede identificar monitoreando estrechamente los puertos TCP/UDP abiertos y el recuento de puertos.
Puertos UDP de escucha	
Recuento de puertos UDP de escucha	

Conceptos

métrica

AWS IoT Device Defender Detect usa métricas para detectar un comportamiento anómalo de los dispositivos. AWS IoT Device Defender Detect compara el valor registrado de una métrica con el valor esperado que proporciona el usuario. Estas métricas se pueden obtener de dos fuentes: de las métricas del lado de la nube y de las métricas del lado del dispositivo: Hay 17 métricas totales, 6 de las cuales son compatibles con ML Detect. Para obtener una lista de las métricas admitidas para ML Detect, consulte [Métricas admitidas \(p. 995\)](#).

El comportamiento anómalo en la red de AWS IoT se detecta mediante el uso de métricas del lado de la nube, como el número de errores de autorización o el número o el tamaño de mensajes que un dispositivo envía o recibe a través de AWS IoT.

AWS IoT Device Defender Detect también puede recopilar, agregar y monitorizar datos de métricas generados por AWS IoT dispositivos (por ejemplo, los puertos que escucha un dispositivo, la cantidad de bytes o paquetes enviados o las conexiones TCP del dispositivo).

Puede usar AWS IoT Device Defender Detect solo con métricas del lado de la nube. Para usar las métricas del lado del dispositivo, primero debe implementar el SDK de AWS IoT en sus dispositivos conectados a AWS IoT o en las gateways de los dispositivos para recopilar las métricas y enviarlas a AWS IoT. Consulte [Envío de métricas desde dispositivos \(p. 1018\)](#).

Perfil de seguridad

Un perfil de seguridad define comportamientos anómalos para un grupo de dispositivos (a [Grupo de objetos \(p. 273\)](#)) o para todos los dispositivos de su cuenta y especifica qué acciones adoptar cuando se detecta una anomalía. Puede utilizar la AWS IoT consola de o API para crear un perfil de seguridad y asociarlo a un grupo de dispositivos. AWS IoT Device Defender Detect comienza a registrar datos relacionados con la seguridad y usa los comportamientos definidos en el Perfil de seguridad para detectar anomalías en el comportamiento de los dispositivos.

comportamiento

Un comportamiento dice AWS IoT Device Defender Detecta cómo puede saber cuándo un dispositivo realiza alguna actividad anómala. Todas las acciones del dispositivo que no coinciden con un comportamiento desencadenan una alerta. Un comportamiento de detección de reglas consiste en

una métrica y un valor absoluto o umbral estadístico con un operador (por ejemplo, menor o igual que, mayor o igual que), que describen el comportamiento esperado del dispositivo. Un comportamiento de detección de ML consiste en una métrica y una configuración de detección de ML, que establecen un modelo ML para conocer el comportamiento normal de los dispositivos.

modelo de ML

Un modelo de ML es un modelo de aprendizaje automático creado para supervisar cada comportamiento que configura un cliente. El modelo entrena en patrones de datos métricos de grupos de dispositivos objetivo y genera tres umbrales de confianza de anomalías (alto, medio y bajo) para el comportamiento basado en métricas. Infiere anomalías basadas en datos métricos ingeridos a nivel de dispositivo. En el contexto de ML Detect, se crea un modelo ML para evaluar un comportamiento basado en métricas. Para obtener más información, consulte [Detección de ML \(p. 993\)](#).

nivel de confianza

ML Detect admite tres niveles de confianza: High, Medium, y Low. High confianza significa baja sensibilidad en la evaluación del comportamiento anómalo y con frecuencia un menor número de alarmas. Medium confianza significa sensibilidad media y Low confianza significa alta sensibilidad y, con frecuencia, un mayor número de alarmas.

dimensión

Puede definir una dimensión para ajustar el ámbito de un comportamiento. Por ejemplo, puede definir una dimensión de filtrado de temas que aplique un comportamiento a los temas de MQTT que coincidan con un patrón. Para obtener más información acerca de cómo definir una dimensión para utilizarla en un perfil de seguridad, consulte [CreateDimension](#).

alarma

Cuando se detecta una anomalía, se puede enviar una notificación de alarma a través de una métrica de CloudWatch (consulte [Uso de las métricas de AWS IoT \(p. 433\)](#)) o una notificación de SNS. También se muestra una notificación de alarma en el AWS IoT Jefe con información adicional sobre la alarma y un historial de las alarmas del dispositivo. También se envía una alarma cuando un dispositivo monitorizado deja de mostrar un comportamiento anómalo o cuando ha estado generando una alarma, pero deja de informar durante un período prolongado.

Verification state

Después de crear una alarma, puede verificar la alarma como Verdadero positivo, positivo benigno, Falso positivo o Desconocido. También puede agregar una descripción del estado de verificación de las alarmas. Puede ver, organizar y filtrar AWS IoT Device Defender alarmas mediante uno de los cuatro estados de verificación. Puedes utilizar los estados de verificación de alarmas y las descripciones relacionadas para informar a los miembros de tu equipo. Esto ayuda a su equipo a realizar acciones de seguimiento, por ejemplo, realizar acciones de mitigación en alarmas positivas verdaderas, omitir alarmas positivas benignas o continuar investigando alarmas desconocidas. El estado de verificación predeterminado para todas las alarmas es Desconocido.

supresión de alarmas

Administrar Detectar notificaciones SNS de alarma configurando la notificación de comportamiento en `not suppressed`. La supresión de alarmas no impide que Detect realice evaluaciones del comportamiento del dispositivo; Detect sigue marcando los comportamientos anómalos como alarmas de infracción. Sin embargo, las alarmas suprimidas no se reenviarían para la notificación de SNS. Solo es posible acceder a ellos a través de la AWS IoT consola o API.

Comportamientos

Un perfil de seguridad contiene un conjunto de comportamientos. Cada comportamiento contiene una métrica que especifica el comportamiento normal de un grupo de dispositivos o de todos los dispositivos de la cuenta. Las conductas se dividen en dos categorías: Reglas Detectar comportamientos

y comportamientos de detección de ML. Con los comportamientos de Rules Detect, define cómo deben comportarse los dispositivos, mientras que ML Detect utiliza modelos ML basados en datos históricos de dispositivos para evaluar cómo deben comportarse los dispositivos.

Un perfil de seguridad puede ser uno de estos dos tipos de umbral: MLoBasado en reglas. Los perfiles de seguridad de ML detectan automáticamente las anomalías operativas y de seguridad a nivel de dispositivo en toda su flota aprendiendo de datos anteriores. Los perfiles de seguridad basados en reglas requieren que establezca manualmente reglas estáticas para supervisar los comportamientos del dispositivo.

A continuación, se describen algunos de los campos que se utilizan en la definición de un behavior:

Común a las reglas Detect y ML Detect

name

El nombre del comportamiento.

metric

El nombre de la métrica utilizada (es decir, lo que mide el comportamiento).

consecutiveDatapointsToAlarm

Si un dispositivo infringe el comportamiento para el número especificado de puntos de datos consecutivos, se genera una alarma. Si no se especifica, el valor predeterminado es 1.

consecutiveDatapointsToClear

Si se genera una alarma y el dispositivo infractor deja de infringir el comportamiento para el número especificado de puntos de datos consecutivos, la alarma se desactiva. Si no se especifica, el valor predeterminado es 1.

threshold type

Un perfil de seguridad puede ser uno de estos dos tipos de umbral: Basado en ML o reglas. Los perfiles de seguridad de ML detectan automáticamente las anomalías operativas y de seguridad a nivel de dispositivo en toda su flota aprendiendo de datos anteriores. Los perfiles de seguridad basados en reglas requieren que establezca manualmente reglas estáticas para supervisar los comportamientos del dispositivo.

alarm suppressions

Administrar Detectar notificaciones SNS de alarma configurando la notificación de comportamiento enonosuppressed. La supresión de alarmas no impide que Detect realice evaluaciones del comportamiento del dispositivo; Detect sigue marcando comportamientos anómalos como alarmas de infracción. Sin embargo, las alarmas suprimidas no se reenvían para la notificación de SNS. Solo se puede acceder a ellos a través de la AWS IoT consola o API.

Detectar reglas

dimension

Puede definir una dimensión para ajustar el ámbito de un comportamiento. Por ejemplo, puede definir una dimensión de filtrado de temas que aplique un comportamiento a los temas de MQTT que coincidan con un patrón. Para definir una dimensión para utilizarla en un perfil de seguridad, consulte [CreateDimension](#). Se aplica únicamente a Rules Detect.

criteria

Los criterios que determinan si un dispositivo se comporta normalmente con respecto a la metric. comparisonOperator

El operador que relaciona el objeto medido (metric) con los criterios (value o statisticalThreshold).

Los valores posibles son: "less-than", "less-than-equals", "greater-than", "greater-than-equals", "in-cidr-set", "not-in-cidr-set", "in-port-set" y "not-in-port-set". No todos los operadores son válidos para todas las métricas. Los operadores para conjuntos y puertos CIDR solo son para usarlos con métricas que impliquen dichas entidades.

value

El valor que se va a comparar con la `metric`. En función del tipo de métrica, debe contener `count` (un valor), `cids` (una lista de CIDR) o `ports` (una lista de puertos).

statisticalThreshold

El umbral estadístico por el que se determina la infracción de un comportamiento. Este campo contiene un campo `statistic` que tiene los siguientes valores posibles: "p0", "p0.1", "p0.01", "p1", "p10", "p50", "p90", "p99", "p99.9", "p99.99" o "p100".

Este campo `statistic` indica un percentil. Da como resultado un valor por el que se determina la conformidad con el comportamiento. Las métricas se recopilan una o más veces durante la duración especificada (`durationSeconds`) de todos los dispositivos de informe asociados a este perfil de seguridad y los percentiles se calculan en función de dichos datos. Posteriormente, las medidas se recopilan para un dispositivo y se acumulan a lo largo de la misma duración. Si el valor resultante del dispositivo está por encima o por debajo (`comparisonOperator`) del valor asociado con el percentil especificado, se considera que el dispositivo se ajusta al comportamiento. De lo contrario, el dispositivo infringirá dicho comportamiento.

Un `percentil` indica el porcentaje de todas las mediciones consideradas que caigan por debajo del valor asociado. Por ejemplo, si el valor asociado a "p90" (el percentil 90.^o) es 123, el 90% de todas las mediciones fueron inferiores a 123.

durationSeconds

Utilícelo para especificar el periodo de tiempo durante el cual se evalúa el comportamiento, para aquellos criterios que tienen una dimensión de tiempo (por ejemplo, `NUM_MESSAGES_SENT`).

Para una comparación de métricas `statisticalThreshold`, se trata del período de tiempo durante el que se recopilan mediciones para todos los dispositivos a fin de determinar los valores `statisticalThreshold` y, a continuación, para cada dispositivo para determinar cómo se compara en comparación.

Detección de ML

ML Detect confidence

ML Detect admite tres niveles de confianza: `High`, `Medium`, y `Low`. `High` confianza significa baja sensibilidad en la evaluación del comportamiento anómalo y con frecuencia un menor número de alarmas. `Medium` confianza significa sensibilidad media, y `Low` confianza significa alta sensibilidad y, con frecuencia, un mayor número de alarmas.

Detección de ML

Con el aprendizaje automático Detect (Detect ML), crea perfiles de seguridad que utilizan el aprendizaje automático para aprender los comportamientos esperados de los dispositivos mediante la creación automática de modelos basados en datos históricos de dispositivos y asigna estos perfiles a un grupo de dispositivos o a todos los dispositivos de su flota. AWS IoT Device Defender identifica las anomalías y activa alarmas utilizando los modelos ML.

Para obtener información sobre cómo empezar a usar ML Detect, consulte [Guía de detección de ML \(p. 885\)](#).

El capítulo contiene las siguientes secciones:

- [Casos de uso de ML Detect \(p. 994\)](#)
- [Funcionamiento de ML Detect \(p. 994\)](#)
- [Requisitos mínimos \(p. 994\)](#)
- [Limitaciones \(p. 995\)](#)
- [Marcar falsos positivos y otros estados de verificación en alarmas \(p. 995\)](#)
- [Métricas admitidas \(p. 995\)](#)
- [Service Quotas \(p. 996\)](#)
- [Comandos de CLI de detección de \(p. 996\)](#)
- [API de detección de ML \(p. 996\)](#)
- [Pausar o eliminar un perfil de seguridad de detección de ML \(p. 996\)](#)

Casos de uso de ML Detect

Puede utilizar ML Detect para supervisar los dispositivos de su flota cuando resulta difícil establecer los comportamientos esperados de los dispositivos. Por ejemplo, para supervisar el número de métricas de desconexiones, puede que no quede claro qué se considera un umbral aceptable. En este caso, puede habilitar ML Detect para identificar puntos de datos de métricas de desconexión anómala en función de los datos históricos notificados desde los dispositivos.

Otro caso de uso de ML Detect consiste en supervisar los comportamientos de los dispositivos que cambian dinámicamente a lo largo del tiempo. ML Detect aprende periódicamente los comportamientos dinámicos esperados de los dispositivos según los patrones de datos cambiantes de los dispositivos. Por ejemplo, el volumen enviado de mensajes del dispositivo podría variar entre días de semana y fines de semana, y la detección de ML aprenderá este comportamiento dinámico.

Funcionamiento de ML Detect

Con ML Detect, puede crear comportamientos para identificar anomalías operativas y de seguridad en todas las[6 métricas de lado de la nube \(p. 995\)](#)y[7 métricas del lado del dispositivo \(p. 995\)](#). Tras el período inicial de formación del modelo, ML Detect actualiza los modelos diariamente en función de los últimos 14 días de datos. Supervisa los puntos de datos de estas métricas con los modelos ML y activa una alarma si se detecta una anomalía.

ML Detect funciona mejor si adjunta un perfil de seguridad a un conjunto de dispositivos con comportamientos esperados similares. Por ejemplo, si algunos de sus dispositivos se utilizan en los hogares de los clientes y otros dispositivos en las oficinas comerciales, los patrones de comportamiento de los dispositivos podrían diferir significativamente entre los dos grupos. Puede organizar los dispositivos en undispositivo de hogargrupo de objetos y undispositivo de oficinagrupo de objetos. Para obtener la mejor eficacia en la detección de anomalías, adjunte cada grupo de cosas a un perfil de seguridad de detección de ML separado.

Mientras ML Detect está creando el modelo inicial, requiere 14 días y un mínimo de 25 000 puntos de datos por métrica durante el período de 14 días finales para generar un modelo. Posteriormente, actualiza el modelo todos los días, hay un número mínimo de puntos de datos métricos. Si no se cumple el requisito mínimo, ML Detect intenta crear el modelo al día siguiente y lo intentará de nuevo diariamente durante los próximos 30 días antes de suspender el modelo para realizar evaluaciones.

Requisitos mínimos

Para la formación y la creación del modelo ML inicial, ML Detect tiene los siguientes requisitos mínimos.

Período de formación mínimo

Los modelos iniciales tardan 14 días en construirse. Despues de eso, el modelo se actualiza todos los días con datos métricos de un período final de 14 días.

Minimum total de puntos de datos

El mínimo de puntos de datos necesarios para crear un modelo de ML es de 25 000 puntos de datos por métrica durante los últimos 14 días. Para la formación continua y la actualización del modelo, ML Detect requiere que se cumplan los puntos de datos mínimos de los dispositivos monitoreados. Es aproximadamente el equivalente a las siguientes configuraciones:

- 60 dispositivos que se conectan y tienen actividad en AWS IoT a intervalos de 45 minutos.
- 40 dispositivos a intervalos de 30 minutos.
- 15 dispositivos a intervalos de 10 minutos.
- 7 dispositivos a intervalos de 5 minutos.

Destinos de grupos de dispositivos

Para que la recopilación de datos progrese, debe tener elementos en los grupos de cosas de destino para el perfil de seguridad.

Una vez creado el modelo inicial, los modelos ML se actualizan todos los días y requieren al menos 25 000 puntos de datos durante el período final de 14 días.

Limitaciones

Actualmente no puedes usar ML Detect con dimensiones o métricas personalizadas. ML Detect no admite las siguientes métricas.

Las métricas del lado de la nube no son compatibles con ML Detect:

- [IP de origen \(aws:dirección-ip de origen\) \(p. 1023\)](#)

Las métricas del lado del dispositivo no son compatibles con ML Detect:

- [IP de destino \(aws:destination-ip-addresses\) \(p. 1010\)](#)
- [Puertos TCP de escucha \(aws:listening-tcp-ports\) \(p. 1010\)](#)
- [Puertos UDP de escucha \(aws:listening-udp-ports\) \(p. 1011\)](#)

Marcar falsos positivos y otros estados de verificación en alarmas

Si verifica que una alarma de detección de ML es falso positivo durante la investigación, puede establecer el estado de verificación de la alarma en Falso positivo. Esto puede ayudarte a ti y a tu equipo a identificar alarmas a las que no tienes que responder. También puedes marcar las alarmas como Verdadero positivo, positivo benigno o Desconocido.

Puedes marcar alarmas a través de la [AWS IoT Device Defender consola](#) mediante el uso de la [Estado de verificación de Put sobre la infracción Acción](#) de la API.

Métricas admitidas

Puedes utilizar las siguientes métricas de lado de la nube con ML Detect:

- [Fallos de autorización \(aws:num-autorización-fallos\) \(p. 1022\)](#)
- [Intentos de conexión \(aws:num-connection-intentos\) \(p. 1024\)](#)
- [Desconexiones \(aws:num-desconexiones\) \(p. 1025\)](#)
- [Tamaño del mensaje \(aws:message-byte-size\) \(p. 1019\)](#)
- [Mensajes enviados \(aws:num-messages-sent\) \(p. 1020\)](#)
- [Mensajes recibidos \(aws:num-messages-recibidos\) \(p. 1021\)](#)

Puedes utilizar las siguientes métricas del lado del dispositivo con la detección de ML:

- Bytes enviados (`aws:all-bytes-out`) (p. 1003)
- Bytes en (`aws:all-bytes-in`) (p. 1004)
- Recuento de puertos TCP de escucha (`aws:num-listening-tcp-ports`) (p. 1005)
- Recuento de puertos UDP de escucha (`aws:num-listening-udp-ports`) (p. 1006)
- Paquetes enviados (`aws:all-packets-out`) (p. 1008)
- Paquetes entrados (`aws:all-packets-in`) (p. 1009)
- Recuento de conexiones TCP establecido (`aws:num-established-tcp-connections`) (p. 1011)

Service Quotas

Para obtener información sobre las cuotas y los límites del servicio ML Detect, consulte [AWS IoT Device Defender Cuotas y puntos de enlace de](#).

Comandos de CLI de detección de

Puede utilizar los siguientes comandos de la CLI de para crear y administrar la detección de ML.

- `create-security-profile`
- `adjunto-seguridad-perfil`
- `list-security-profiles`
- `describe-security-profile`
- `update-security-profile`
- `delete-security-profile`
- `obtener resúmenes de formación de modelo-comportamiento-modelos-`
- `lista-activo-infracciones`
- `violación-lista-sucesos`

API de detección de ML

Las siguientes API se pueden utilizar para crear y administrar perfiles de seguridad de detección de ML.

- `CreateSecurityProfile`
- `AttachSecurityProfile`
- `ListSecurityProfiles`
- `DescribeSecurityProfile`
- `UpdateSecurityProfile`
- `DeleteSecurityProfile`
- `Obtener resúmenes de formación del modelo de comportamiento`
- `ListActiveViolations`
- `ListViolationEvents`
- Estado de verificación de Put sobre la infracción

Pausar o eliminar un perfil de seguridad de detección de ML

Puede pausar el perfil de seguridad de detección de ML para dejar de supervisar temporalmente los comportamientos del dispositivo o eliminar el perfil de seguridad de detección de ML para dejar de supervisar los comportamientos de los dispositivos durante un período prolongado.

Pausa ML Detect Security Profile mediante la consola

Para pausar un perfil de seguridad de detección de ML mediante la consola, primero debe tener un grupo de cosas vacío. Para crear un grupo de objetos vacío, consulte[Grupos de objetos estáticos \(p. 273\)](#). Si ha creado un grupo de cosas vacío, establezca el grupo de cosas vacío como destino del perfil de seguridad de detección de ML.

Note

Debe volver a configurar el objetivo de su perfil de seguridad en un grupo de dispositivos con dispositivos en un plazo de 30 días o no podrá reactivar el perfil de seguridad.

Eliminar perfil de seguridad de detección de ML mediante la consola

Para eliminar un perfil de seguridad, sigue estos pasos:

1. En el navegadorAWS IoTconsola navegue hasta la barra lateral y elija laDefendersección.
2. UnderDefender, eligeDetectary luegoProfiles de seguridad de.
3. Elija el perfil de seguridad de detección de ML que desea eliminar.
4. ElegirActionsy, a continuación, en las opciones, elijaBorrar.

Note

Una vez eliminado un perfil de seguridad de detección de ML, no podrá reactivar el perfil de seguridad.

Pausa un perfil de seguridad de detección de ML mediante la CLI

Para pausar un perfil de seguridad de detección de ML mediante la CLI, utilice eldetach-security-profilecomando:

```
$aws iot detach-security-profile --security-profile-name SecurityProfileName --  
security-profile-target-arn arn:aws:iot:us-east-1:123456789012:all/registered-things
```

Note

Esta opción solo está disponible enAWSCLI. Al igual que en el flujo de trabajo de la consola, debe volver a configurar el objetivo de su perfil de seguridad en un grupo de dispositivos con dispositivos en un plazo de 30 días o no podrá reactivar el perfil de seguridad. Para adjuntar un perfil de seguridad a un grupo de dispositivos, utilice elattach-security-profilecomando.

Eliminar un perfil de seguridad de detección de ML mediante la CLI

Puede eliminar un perfil de seguridad mediante eldelete-security-profilecomando a continuación:

```
delete-security-profile --security-profile-name SecurityProfileName
```

Note

Una vez eliminado un perfil de seguridad de detección de ML, no podrá reactivar el perfil de seguridad.

Métricas de personalizadas

conAWS IoT Device Defendermétricas personalizadas, puede definir y supervisar métricas exclusivas de su flota o caso de uso, como el número de dispositivos conectados a las puertas de enlace Wi-Fi, los niveles de carga de las baterías o el número de ciclos de alimentación para enchufes inteligentes.

Los comportamientos de métrica personalizados se definen en Perfiles de seguridad, que especifican comportamientos esperados para un grupo de dispositivos (un grupo de elementos) o para todos los dispositivos. Puede supervisar los comportamientos configurando alarmas, que puede utilizar para detectar y responder a problemas específicos de los dispositivos.

El capítulo contiene las siguientes secciones:

- [Uso de métricas personalizadas en la consola \(p. 998\)](#)
- [Uso de métricas personalizadas de la CLI \(p. 1000\)](#)
- [Métricas personalizadas de comandos de la CLI \(p. 1002\)](#)
- [API de métricas personalizadas \(p. 1003\)](#)

Uso de métricas personalizadas en la consola

Tutoriales

- [AWS IoT Device DefenderSDK de agente \(Python\) \(p. 998\)](#)
- [Crear una métrica personalizada y añadirla a un perfil de seguridad \(p. 998\)](#)
- [Visualizar detalles de métricas personalizadas \(p. 999\)](#)
- [Actualización de una métrica personalizada \(p. 999\)](#)
- [Eliminación de una métrica personalizada \(p. 999\)](#)

AWS IoT Device DefenderSDK de agente (Python)

Para empezar, descargue AWS IoT Device DefenderAgente de ejemplo del SDK del agente (Python). El agente recopila las métricas y publica informes. Una vez publicadas las métricas del lado del dispositivo, puedes ver las métricas que se recopilan y determinar los umbrales para configurar alarmas. Las instrucciones para configurar el agente de dispositivos están disponibles en el[AWS IoT Léame del SDK del agente Device Defender \(Python\)](#). Para obtener más información, consulte[AWS IoT Device DefenderSDK de agente \(Python\)](#).

Crear una métrica personalizada y añadirla a un perfil de seguridad

En el siguiente procedimiento se muestra cómo crear una métrica personalizada en la consola de.

1. En el navegador[AWS IoT consola](#), en el panel de navegación, expandaDefendery luego enDetectar,Métricas.
2. En la páginaMétricas personalizadaspágina, elijaCrear.
3. En la páginaCreación de una métrica personalizadaapágina, haga lo siguiente:
 1. UNDERNombre, escriba un nombre para la métrica personalizada. Este nombre no se puede modificar después de haber creado la métrica personalizada.
 2. UNDERDisplay name (Nombre que mostrar) (opcional), puede introducir un nombre fácil de leer para su métrica personalizada. No tiene que ser único y se puede modificar después de su creación.
 3. UNDERTipo, elige el tipo de métrica que quieras supervisar. Los tipos de métricas incluyenlista de cadenas,lista de direcciones IP,lista numérica, ynúmero. El tipo no se puede modificar después de la creación.
 4. UNDEREtiquetas, puede seleccionar las etiquetas para asociarlas con el recurso.

Cuando haya terminado, elijaConfirmar.

4. Despues de crear la métrica personalizada, elMétricas personalizadasAparece una página en la que podrá ver la métrica personalizada recién creada.

5. A continuación, debe añadir la métrica personalizada a un perfil de seguridad. En el navegador[AWS IoTconsola](#), en el panel de navegación, expandaDefendery luego enDetectar,Perfiles de seguridad.
6. Elije el perfil de seguridad al que quieres añadir tu métrica personalizada.
7. Elija Actions, Edit.
8. ElegirMétricas adicionales que se deben conservary, a continuación, elija su métrica personalizada. ElegirPróximoen las siguientes pantallas hasta que llegues alConfirmar(Se ha creado el certificado). ElegirGuardaryContinuar. Una vez que la métrica personalizada se haya agregado correctamente, aparece la página de detalles del perfil de seguridad.

Note

Las estadísticas de percentiles no están disponibles para las métricas cuando alguno de los valores de métricas es un número negativo.

Visualizar detalles de métricas personalizadas

En el siguiente procedimiento se muestra cómo ver los detalles de una métrica personalizada en la consola de.

1. En el navegador[AWS IoTconsola](#), en el panel de navegación, expandaDefendery luego enDetectar,Métricas.
2. Elija el iconoNombre de métricade la métrica personalizada de la que quieres ver los detalles.

Actualización de una métrica personalizada

En el siguiente procedimiento se muestra cómo actualizar una métrica personalizada en la consola de.

1. En el navegador[AWS IoTconsola](#), en el panel de navegación, expandaDefendery luego enDetectar,Métricas.
2. Elija el botón de opción situado junto a la métrica personalizada que desea actualizar. Luego, paraActions, eligeEditar.
3. En la páginaActualización de la métrica personalizada, puede editar el nombre para mostrar y eliminar o añadir etiquetas.
4. Una vez que haya terminado, elijaActualización. LaMétricas personalizadas(Se ha creado el certificado).

Eliminación de una métrica personalizada

En el siguiente procedimiento se muestra cómo eliminar una métrica personalizada de la consola de.

1. En primer lugar, elimina la métrica personalizada de cualquier perfil de seguridad en el que se haga referencia. Puede ver qué perfiles de seguridad contienen la métrica personalizada en la página de detalles de métricas personalizadas. En el navegador[AWS IoTconsola](#), en el panel de navegación, expandaDefendery luego enDetectar,Métricas.
2. Elija la métrica personalizada que desea eliminar. Eliminar la métrica personalizada de cualquier perfil de seguridad que aparece enPerfiles de seguridad deen la página de detalles de métricas personalizadas.
3. En el navegador[AWS IoTconsola](#), en el panel de navegación, expandaDefendery luego enDetectar,Métricas.
4. Elija el botón de opción situado junto a la métrica personalizada que desea eliminar. Luego, paraActions, eligeBorrar.
5. En la página¿Seguro que desea eliminar la métrica personalizada?mensaje, elijaEliminación de métrica personalizada.

Warning

Después de eliminar una métrica personalizada, pierde todos los datos asociados a la métrica. Esta acción no se puede deshacer.

Uso de métricas personalizadas de la CLI

Tutoriales

- [AWS IoT Device DefenderSDK de agente \(Python\) \(p. 1000\)](#)
- [Crear una métrica personalizada y añadirla a un perfil de seguridad \(p. 1000\)](#)
- [Visualizar detalles de métricas personalizadas \(p. 1001\)](#)
- [Actualización de una métrica personalizada \(p. 1001\)](#)
- [Eliminación de una métrica personalizada \(p. 1002\)](#)

AWS IoT Device DefenderSDK de agente (Python)

Para empezar, descargue [AWS IoT Device DefenderAgente de ejemplo del SDK del agente \(Python\)](#). El agente recopila las métricas y publica informes. Una vez publicadas las métricas del lado del dispositivo, puedes ver las métricas que se recopilan y determinar los umbrales para configurar alarmas. Las instrucciones para configurar el agente de dispositivos están disponibles en el [AWS IoT Léame del SDK del agente Device Defender \(Python\)](#). Para obtener más información, consulte [AWS IoT Device DefenderSDK de agente \(Python\)](#).

Crear una métrica personalizada y añadirla a un perfil de seguridad

En el siguiente procedimiento se muestra cómo crear una métrica personalizada y agregarla a un perfil de seguridad desde la CLI.

1. Usar `create-custom-metric` para crear la métrica personalizada. En el siguiente ejemplo se crea una métrica personalizada que mide el porcentaje de batería.

```
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" \
```

Salida:

```
{ \
    "metricName": "batteryPercentage", \
    "metricArn": "arn:aws:iot:us-east-1:1234564789012:custommetric/batteryPercentage" \
}
```

2. Despues de crear la métrica personalizada, puede agregar la métrica personalizada a un perfil existente mediante `update-security-profile` o bien, cree un perfil de seguridad para añadir la métrica personalizada a `create-security-profile`. A continuación, creamos un perfil de seguridad llamado [Uso de batería](#) para añadir nuestro nuevo [Porcentaje de batería](#) métricas personalizadas a. También agregamos una métrica Rules Detect llamada [Ancho 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
```

Salida:

```
{
    "securityProfileArn": "arn:aws:iot:us-east-1:1234564789012:securityprofile/
batteryUsage",
    "securityProfileName": "batteryUsage"
}
```

Note

Las estadísticas de percentiles no están disponibles para las métricas cuando alguno de los valores de métricas es un número negativo.

Visualizar detalles de métricas personalizadas

En el siguiente procedimiento se muestra cómo ver los detalles de una métrica personalizada desde la CLI.

- Usar[list-custom-metrics](#)para ver todas las métricas personalizadas.

```
aws iot list-custom-metrics \
--region us-east-1
```

El resultado de este comando tendrá un aspecto similar al siguiente.

```
{
    "metricNames": [
        "batteryPercentage"
    ]
}
```

Actualización de una métrica personalizada

En el siguiente procedimiento se muestra cómo actualizar una métrica personalizada desde la CLI.

- Usar[update-custom-metric](#)para actualizar una métrica personalizada. En el ejemplo siguiente se actualiza `display-name`.

```
aws iot update-custom-metric \
--metric-name batteryPercentage \
--display-name 'remaining battery percentage on device' \
--region us-east-1
```

El resultado de este comando tendrá un aspecto similar al siguiente.

```
{
    "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"
    }
```

Eliminación de una métrica personalizada

En el siguiente procedimiento se muestra cómo eliminar una métrica personalizada de la CLI.

1. Para eliminar una métrica personalizada, primero quítela de los perfiles de seguridad a los que esté asociada. Usar [list-security-profiles](#) para ver los perfiles de seguridad con una determinada métrica personalizada.
2. Para quitar una métrica personalizada de un perfil de seguridad, utilice la [update-security-profiles](#) comando. Introduzca toda la información que desee conservar, pero excluya la 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}}]"
```

El resultado de este comando tendrá un aspecto similar al siguiente.

```
{  
    "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. Una vez desconectada la métrica personalizada, utilice [delete-custom-metric](#) para eliminar la métrica personalizada.

```
aws iot delete-custom-metric \
--metric-name batteryPercentage \
--region us-east-1
```

El resultado de este comando tendrá un aspecto similar al siguiente.

```
HTTP 200
```

Métricas personalizadas de comandos de la CLI

Puede utilizar los siguientes comandos de la CLI para crear y administrar métricas personalizadas.

- [create-custom-metric](#)
- [describe-métrico-personalizado](#)

- [list-custom-metrics](#)
- [actualización-métrica personalizada](#)
- [borrar métrica personalizada](#)
- [list-security-profiles](#)

API de métricas personalizadas

Las siguientes API se pueden utilizar para crear y administrar métricas personalizadas.

- [Crear métrica personalizada](#)
- [Descripción Custom Metric](#)
- [Listar métricas personalizadas](#)
- [Actualizar métrica personalizada](#)
- [Eliminar métrica personalizada](#)
- [ListSecurityProfiles](#)

Métricas del lado del dispositivo

Al crear un perfil de seguridad, puede especificar el comportamiento esperado de su dispositivo IoT configurando comportamientos y umbrales para las métricas generadas por los dispositivos IoT. Las siguientes son métricas del lado del dispositivo, que son métricas de agentes instalados en los dispositivos.

Bytes enviados (aws:all-bytes-out)

El número de bytes salientes de un dispositivo durante un período de tiempo determinado.

Utilice esta métrica para especificar la cantidad máxima o mínima de tráfico saliente que un dispositivo debe enviar, medido en bytes, en un período de tiempo determinado.

Compatible con: Detección de reglas | ML Detect

Operadores: less-than | less-than-equals | greater-than | greater-than-equals

Valor: un entero no negativo

Unidades: bytes

Duración: un entero no negativo. Los valores válidos son 300, 600, 900, 1800 o 3600 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 Ejemplo en el que se utiliza **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 Ejemplo de uso de ML Detect

```
{
  "name": "Outbound traffic ML behavior",
  "metric": "aws:all-bytes-out",
  "criteria": {
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1,
    "mlDetectionConfig": {
      "confidenceLevel": "HIGH"
    }
  },
  "suppressAlerts": true
}
```

Bytes en (aws:all-bytes-in)

El número de bytes entrantes en un dispositivo durante un período de tiempo determinado.

Utilice esta métrica para especificar la cantidad máxima o mínima de tráfico entrante que un dispositivo debe recibir, medido en bytes, en un período de tiempo determinado.

Compatible con: Detección de reglas | ML Detect

Operadores: less-than | less-than-equals | greater-than | greater-than-equals

Valor: un entero no negativo

Unidades: bytes

Duración: un entero no negativo. Los valores válidos son 300, 600, 900, 1800 o 3600 segundos.

Example

```
{
  "name": "TCP inbound traffic",
  "metric": "aws:all-bytes-in",
  "criteria": {
    "comparisonOperator": "less-than-equals",
    "durationSeconds": 900
  }
}
```

```
        "value": {
            "count": 4096
        },
        "durationSeconds": 300,
        "consecutiveDatapointsToAlarm": 1,
        "consecutiveDatapointsToClear": 1
    },
    "suppressAlerts": true
}
```

Example Ejemplo en el que se utiliza **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 Ejemplo de uso de ML Detect

```
{
    "name": "Inbound traffic ML behavior",
    "metric": "aws:all-bytes-in",
    "criteria": {
        "consecutiveDatapointsToAlarm": 1,
        "consecutiveDatapointsToClear": 1,
        "mlDetectionConfig": {
            "confidenceLevel": "HIGH"
        }
    },
    "suppressAlerts": true
}
```

Recuento de puertos TCP de escucha (aws: num-listening-tcp-ports)

El número de puertos TCP en los que escucha el dispositivo.

Utilice esta métrica para especificar el número máximo de puertos TCP que cada dispositivo debe supervisar.

Compatible con: Detección de reglas | ML Detect

Unidad: errores

Operadores: less-than | less-than-equals | greater-than | greater-than-equals

Valor: un entero no negativo

Unidades: errores

Duración: un entero no negativo. Los valores válidos son 300, 600, 900, 1800 o 3600 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 Ejemplo en el que se utiliza **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 Ejemplo de uso de detección de ML

```
{  
  "name": "Max TCP Port ML behavior",  
  "metric": "aws:num-listening-tcp-ports",  
  "criteria": {  
    "consecutiveDatapointsToAlarm": 1,  
    "consecutiveDatapointsToClear": 1,  
    "mlDetectionConfig": {  
      "confidenceLevel": "HIGH"  
    }  
  },  
  "suppressAlerts": true  
}
```

Recuento de puertos UDP de escucha (aws:num-listening-udp-ports)

El número de puertos UDP en los que escucha el dispositivo.

Utilice esta métrica para especificar el número máximo de puertos UDP que cada dispositivo debe supervisar.

Compatible con: Detección de reglas | ML Detect

Unidad: errores

Operadores: less-than | less-than-equals | greater-than | greater-than-equals

Valor: un entero no negativo

Unidades: errores

Duración: un entero no negativo. Los valores válidos son 300, 600, 900, 1800 o 3600 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 Ejemplo en el que se utiliza **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 Ejemplo de uso de ML Detect

```
{  
  "name": "Max UPD Port ML behavior",  
  "metric": "aws:num-listening-tcp-ports",  
  "criteria": {  
    "consecutiveDatapointsToAlarm": 1,  
    "consecutiveDatapointsToClear": 1,  
    "mlDetectionConfig": {  
      "confidenceLevel": "HIGH"  
    },  
    "suppressAlerts": true  
  }
```

Paquetes enviados (aws:all-packets-out)

El número de paquetes salientes de un dispositivo durante un período de tiempo determinado.

Utilice esta métrica para especificar la cantidad máxima o mínima de tráfico saliente total que un dispositivo debe enviar en un período de tiempo determinado.

Compatible con: Detección de reglas | ML Detect

Operadores: less-than | less-than-equals | greater-than | greater-than-equals

Valor: un entero no negativo

Unidades: paquetes

Duración: un entero no negativo. Los valores válidos son 300, 600, 900, 1800 o 3600 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 Ejemplo en el que se utiliza **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 Ejemplo de uso de ML Detect

```
{  
  "name": "Outbound sent ML behavior",  
  "metric": "aws:all-packets-out",  
  "criteria": {  
    "consecutiveDatapointsToAlarm": 1,  
    "consecutiveDatapointsToClear": 1,  
  }
```

```
    "mlDetectionConfig": {
        "confidenceLevel": "HIGH"
    },
    "suppressAlerts": true
}
```

Paquetes entrados (aws:all-packets-in)

El número de paquetes entrantes en un dispositivo durante un período de tiempo determinado.

Utilice esta métrica para especificar la cantidad máxima o mínima de tráfico entrante total que un dispositivo debe recibir en un período de tiempo determinado.

Compatible con: Detección de reglas | ML Detect

Operadores: less-than | less-than-equals | greater-than | greater-than-equals

Valor: un entero no negativo

Unidades: paquetes

Duración: un entero no negativo. Los valores válidos son 300, 600, 900, 1800 o 3600 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

Ejemplo en el que se utiliza 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 Ejemplo de uso de ML Detect

```
{  
    "name": "Inbound sent ML behavior",  
    "metric": "aws:all-packets-in",  
    "criteria": {  
        "consecutiveDatapointsToAlarm": 1,  
        "consecutiveDatapointsToClear": 1,  
        "mlDetectionConfig": {  
            "confidenceLevel": "HIGH"  
        }  
    },  
    "suppressAlerts": true  
}
```

IP de destino (aws:destination-ip-addresses)

Un conjunto de destinos de IP.

Utilice esta métrica para especificar un conjunto de enrutes entre dominios sin clase (CIDR) y denegados (previamente denominado lista negra) a los que cada dispositivo debe o no conectarse aAWS IoT.

Compatible con: Reglas detectadas

Operadores: in-cidr-set | not-in-cidr-set

Valores: una lista de CIDR

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  
}
```

Puertos TCP de escucha (aws:listening-tcp-ports)

Los puertos TCP en los que escucha el dispositivo.

Utilice esta métrica para especificar un conjunto de puertos TCP permitidos (previamente denominado lista blanca) y denegados (previamente denominado lista negra) en los que cada dispositivo debe o no escuchar.

Compatible con: Reglas detectadas

Operadores: in-port-set | not-in-port-set

Valores: una lista de puertos

Unidades: n/a

Example

```
{  
    "name": "Listening TCP Ports",  
    "metric": "aws:listening-tcp-ports",  
    "criteria": {  
        "comparisonOperator": "in-port-set",  
        "value": {  
            "ports": [ 443, 80 ]  
        }  
    },  
    "suppressAlerts": true  
}
```

Puertos UDP de escucha (aws:listening-udp-ports)

Los puertos UDP en los que escucha el dispositivo.

Utilice esta métrica para especificar un conjunto de puertos UDP permitidos (previamente denominado lista blanca) y denegados (previamente denominado lista negra) en los que cada dispositivo debe o no escuchar.

Compatible con: Reglas detectadas

Operadores: in-port-set | not-in-port-set

Valores: una lista de puertos

Unidades: n/a

Example

```
{  
    "name": "Listening UDP Ports",  
    "metric": "aws:listening-udp-ports",  
    "criteria": {  
        "comparisonOperator": "in-port-set",  
        "value": {  
            "ports": [ 1025, 2000 ]  
        }  
    }  
}
```

Recuento de conexiones TCP establecido (aws:num-established-tcp-connections)

El número de conexiones TCP para un dispositivo.

Utilice esta métrica para especificar el número máximo o mínimo de conexiones TCP activas que cada dispositivo debe tener.

Compatible con: Detección de reglas | ML Detect

Operadores: less-than | less-than-equals | greater-than | greater-than-equals

Valor: un entero no negativo

Unidades: conexiones

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 Ejemplo en el que se utiliza **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 Ejemplo de uso de ML Detect

```
{  
    "name": "Connection count ML behavior",  
    "metric": "aws:num-established-tcp-connections",  
    "criteria": {  
        "consecutiveDatapointsToAlarm": 1,  
        "consecutiveDatapointsToClear": 1,  
        "mlDetectionConfig": {  
            "confidenceLevel": "HIGH"  
        }  
    },  
    "suppressAlerts": true  
}
```

Especificación de documentos de métricas de dispositivos

Estructura general

Nombre largo	Nombre corto	Obligatorio	Tipo	Restricciones	Notas
header	hed	S	Objeto		Bloque completo obligatorio para componer un informe correcto.

Nombre largo	Nombre corto	Obligatorio	Tipo	Restricciones	Notas
métricas	met	S	Objeto		Un informe puede tener ambos o al menos un <code>metrics</code> o <code>custom_metrics</code> .
custom_metrics	cmet	S	Objeto		Un informe puede tener ambos o al menos un <code>metrics</code> o <code>custom_metrics</code> .

Bloque de encabezado

Nombre largo	Nombre corto	Obligatorio	Tipo	Restricciones	Notas
report_id	rid	S	Entero		Valor monótonico en aumento. Se recomienda una marca temporal Epoch.
version	v	S	Cadena	Major.Minor	Incrementos de versiones secundarias con suma de campo. Incrementos de versiones principales si se eliminan las métricas.

Bloque de métricas:

Conexiones de TCP

Nombre largo	Nombre corto	Elemento principal	Obligatorio	Tipo	Restricciones	Notas
tcp_connection\$c		métricas	N	Objeto		
established_connections		tcp_connections\$N		Objeto		Estado de TCP establecido
connections	cs	established_connections	N	Lista<Objeto>		
remote_addr	rad	connections	S	Número	ip:port	La IP puede ser IPv6 o IPv4
local_port	lp	connections	N	Número	>= 0	

Nombre largo	Nombre corto	Elemento principal	Obligatorio	Tipo	Restricciones	Notas
local_interface	li	connections	N	Cadena		Nombre de la interfaz
total	t	established_connections	N	Número	>= 0	Número de conexiones establecidas

Puertos TCP de escucha

Nombre largo	Nombre corto	Elemento principal	Obligatorio	Tipo	Restricciones	Notas
listening_tcp_ports	pts	métricas	N	Objeto		
ports	pts	listening_tcp_ports	N	Lista<Objeto>	> 0	
port	pt	ports	N	Número	> 0	Los puertos deben ser números mayores que 0
interface	if	ports	N	Cadena		Nombre de la interfaz
total	t	listening_tcp_ports	N	Número	>= 0	

Puertos UDP de escucha

Nombre largo	Nombre corto	Elemento principal	Obligatorio	Tipo	Restricciones	Notas
listening_udp_ports	pts	métricas	N	Objeto		
ports	pts	listening_udp_ports	N	Lista<Puerto>	> 0	
port	pt	ports	N	Número	> 0	Los puertos deben ser números mayores que 0
interface	if	ports	N	Cadena		Nombre de la interfaz
total	t	listening_udp_ports	N	Número	>= 0	

Estadísticas de la red

Nombre largo	Nombre corto	Elemento principal	Obligatorio	Tipo	Restricciones	Notas
network_stats	ns	métricas	N	Objeto		

Nombre largo	Nombre corto	Elemento principal	Obligatorio	Tipo	Restricciones	Notas
bytes_in	bi	network_stats	N	Número	Delta Metric, ≥ 0	
bytes_out	bo	network_stats	N	Número	Delta Metric, ≥ 0	
packets_in	pi	network_stats	N	Número	Delta Metric, ≥ 0	
packets_out	po	network_stats	N	Número	Delta Metric, ≥ 0	

Example

La siguiente estructura JSON utiliza nombres largos.

```
{
  "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 Ejemplo de una estructura JSON con nombres cortos

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

Envío de métricas desde dispositivos

AWS IoT Device Defender Detect puede recopilar, agregar y monitorear los datos de métricas generados por los dispositivos de AWS IoT para identificar aquellos dispositivos que muestran un comportamiento anómalo. En esta sección, se explica cómo enviar métricas desde un dispositivo a AWS IoT Device Defender.

Debe desplegar de forma segura elAWS IoTSDK versión dos en suAWS IoTdispositivos conectados o puertas de enlace de dispositivos para recopilar métricas del lado del dispositivo. Consulte la lista completa de SDK[aquí](#).

Puede usarAWS IoTDevice Client para publicar métricas, ya que proporciona un único agente que cubre las características presentes en ambosAWS IoT Device DefenderyAWS IoTAdministración de dispositivos. Estas funciones incluyen trabajos, túneles seguros,AWS IoT Device Defenderpublicación de métricas y mucho más.

Publica métricas del lado del dispositivo en el[Tema reservado](#)enAWS IoTparaAWS IoT Device Defenderrecopilar y evaluar.

Uso deAWS IoTDevice Client para publicar métricas

Para instalarAWS IoTDevice Client, puede descargarlo desde[Github](#). Después de instalar elAWS IoTDevice Client en el dispositivo para el que desea recopilar datos del lado del dispositivo, debe configurarlo para enviar métricas del lado del dispositivo aAWS IoT Device Defender. Verifique que laAWS IoTCliente del dispositivo[Archivo de configuración](#)tiene los parámetros siguientes configurados en eldevice-defendersección :

```
"device-defender": {
    "enabled": true,
    "interval-in-seconds": 300
}
```

Warning

Debe establecer el intervalo de tiempo en un mínimo de 300 segundos. Si establece el intervalo de tiempo en menos de 300 segundos, es posible que los datos de las métricas se reduzcan.

Después de actualizar la configuración, puede crear perfiles y comportamientos de seguridad en elAWS IoT Device Defenderconsola para supervisar las métricas que publican sus dispositivos en la nube. Puedes encontrar métricas publicadas en elAWS IoT Coreeligiendo Defender, Detectar y, a continuación, Métricas.

Métricas del lado de la nube

Al crear un perfil de seguridad, puede especificar el comportamiento esperado de su dispositivo IoT configurando comportamientos y umbrales para las métricas generadas por los dispositivos IoT. Las siguientes son métricas del lado de la nube, que son métricas de AWS IoT.

Tamaño del mensaje (aws:message-byte-size)

El número de bytes de un mensaje. Utilice esta métrica para especificar el tamaño máximo o mínimo (en bytes) de cada mensaje transmitido desde un dispositivo a AWS IoT.

Compatible con: Detección de reglas | ML Detect

Operadores: less-than | less-than-equals | greater-than | greater-than-equals

Valor: un entero no negativo

Unidades: 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 Ejemplo en el que se utiliza **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 Ejemplo de uso de ML Detect

```
{  
    "name": "Message size ML behavior",  
    "metric": "aws:message-byte-size",  
    "criteria": {
```

```
"consecutiveDatapointsToAlarm": 1,  
"consecutiveDatapointsToClear": 1,  
"mlDetectionConfig": {  
    "confidenceLevel": "HIGH"  
},  
"suppressAlerts": true  
}
```

Se genera una alarma para un dispositivo si durante tres períodos consecutivos de cinco minutos transmite mensajes en los que el tamaño acumulado es mayor que el medido para el 90 por ciento de todos los demás dispositivos que informan sobre este comportamiento de perfil de seguridad.

Mensajes enviados (aws:num-messages-sent)

El número de mensajes enviados por un dispositivo durante un período de tiempo determinado.

Utilice esta métrica para especificar el número máximo o mínimo de mensajes que pueden enviarse entre AWS IoT cada dispositivo en un período de tiempo determinado.

Compatible con: Detección de reglas | ML Detect

Operadores: less-than | less-than-equals | greater-than | greater-than-equals

Valor: un entero no negativo

Unidades: mensajes

Duración: un entero no negativo. Los valores válidos son 300, 600, 900, 1800 o 3600 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 Ejemplo en el que se utiliza **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  
}
```

```
        "consecutiveDatapointsToClear": 1
    },
    "suppressAlerts": true
}
```

Example Ejemplo de uso de ML Detect

```
{
  "name": "Messages sent ML behavior",
  "metric": "aws:num-messages-sent",
  "criteria": {
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1,
    "mlDetectionConfig": {
      "confidenceLevel": "HIGH"
    }
  },
  "suppressAlerts": true
}
```

Mensajes recibidos (aws:num-messages-recibidos)

El número de mensajes recibidos por un dispositivo durante un período de tiempo determinado.

Utilice esta métrica para especificar el número máximo o mínimo de mensajes que pueden recibirse entre AWS IoT cada dispositivo en un período de tiempo determinado.

Compatible con: Detección de reglas | ML Detect

Operadores: less-than | less-than-equals | greater-than | greater-than-equals

Valor: un entero no negativo

Unidades: mensajes

Duración: un entero no negativo. Los valores válidos son 300, 600, 900, 1800 o 3600 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 Ejemplo en el que se utiliza **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 Ejemplo de uso de ML Detect

```
{  
    "name": "Messages received ML behavior",  
    "metric": "aws:num-messages-received",  
    "criteria": {  
        "consecutiveDatapointsToAlarm": 1,  
        "consecutiveDatapointsToClear": 1,  
        "mlDetectionConfig": {  
            "confidenceLevel": "HIGH"  
        }  
    },  
    "suppressAlerts": true  
}
```

Fallos de autorización (aws:num-autorización-fallos)

Utilice esta métrica para especificar la cantidad máxima de errores de autorización permitidos para cada dispositivo en un período de tiempo determinado. Se produce un error de autorización cuando se deniega una solicitud desde un dispositivo a AWS IoT (por ejemplo, si un dispositivo intenta publicar en un tema para el que no tiene suficientes permisos).

Compatible con: Detección de reglas | ML Detect

Unidad: errores

Operadores: less-than | less-than-equals | greater-than | greater-than-equals

Valor: un entero no negativo

Duración: un entero no negativo. Los valores válidos son 300, 600, 900, 1800 o 3600 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 Ejemplo en el que se utiliza **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 Ejemplo de uso de ML Detect

```
{  
    "name": "Authorization failures ML behavior",  
    "metric": "aws:num-authorization-failures",  
    "criteria": {  
        "consecutiveDatapointsToAlarm": 1,  
        "consecutiveDatapointsToClear": 1,  
        "mlDetectionConfig": {  
            "confidenceLevel": "HIGH"  
        },  
        "suppressAlerts": true  
    }
```

IP de origen (aws:dirección-ip de origen)

La dirección IP desde la que se ha conectado un dispositivo a AWS IoT.

Utilice esta métrica para especificar un conjunto de enrutamientos entre dominios sin clase (previamente denominado lista blanca) y denegados (previamente denominado lista negra) y denegados (previamente denominado lista negra) desde los que cada dispositivo debe o no conectarse aAWS IoT.

Compatible con: Detectar reglas

Operadores: in-cidr-set | not-in-cidr-set

Valores: una lista de CIDR

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  
}
```

}

Intentos de conexión (aws:num-connection-intentos)

Número de veces que un dispositivo intenta realizar una conexión durante un periodo determinado.

Utilice esta métrica para especificar el número máximo o mínimo de intentos de conexión para cada dispositivo. Se cuentan los intentos correctos y los no correctos.

Compatible con: Detección de reglas | ML Detect

Operadores: less-than | less-than-equals | greater-than | greater-than-equals

Valor: un entero no negativo

Unidades: intentos de conexión

Duración: un entero no negativo. Los valores válidos son 300, 600, 900, 1800 o 3600 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 Ejemplo en el que se utiliza **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 Ejemplo de uso de ML Detect

```
{  
  "name": "Connection attempts ML behavior",  
  "metric": "aws:num-connection-attempts",  
  "criteria": {
```

```
        "consecutiveDatapointsToAlarm": 1,
        "consecutiveDatapointsToClear": 1,
        "mlDetectionConfig": {
            "confidenceLevel": "HIGH"
        }
    },
    "suppressAlerts": false
}
```

Desconexiones (aws:num-desconexiones)

Número de veces que un dispositivo se desconecta de AWS IoT durante un periodo de tiempo determinado.

Utilice esta métrica para especificar el número máximo o mínimo de veces que un dispositivo se ha desconectado de AWS IoT durante un periodo de tiempo determinado.

Compatible con: Detección de reglas | ML Detect

Operadores: less-than | less-than-equals | greater-than | greater-than-equals

Valor: un entero no negativo

Unidades: desconexiones

Duración: un entero no negativo. Los valores válidos son 300, 600, 900, 1800 o 3600 segundos.

Example

```
{
    "name": "Disconnects",
    "metric": "aws:num-disconnects",
    "criteria": {
        "comparisonOperator": "less-than-equals",
        "value": {
            "count": 5
        },
        "durationSeconds": 600,
        "consecutiveDatapointsToAlarm": 1,
        "consecutiveDatapointsToClear": 1
    },
    "suppressAlerts": true
}
```

Example Ejemplo en el que se utiliza **statisticalThreshold**:

```
{
    "name": "Disconnects",
    "metric": "aws:num-disconnects",
    "criteria": {
        "comparisonOperator": "less-than-equals",
        "statisticalThreshold": {
            "statistic": "p10"
        },
        "durationSeconds": 300,
        "consecutiveDatapointsToAlarm": 1,
        "consecutiveDatapointsToClear": 1
    },
    "suppressAlerts": true
}
```

Example Ejemplo de uso de ML Detect

```
{  
    "name": "Disconnects ML behavior",  
    "metric": "aws:num-disconnects",  
    "criteria": {  
        "consecutiveDatapointsToAlarm": 1,  
        "consecutiveDatapointsToClear": 1,  
        "mlDetectionConfig": {  
            "confidenceLevel": "HIGH"  
        }  
    },  
    "suppressAlerts": true  
}
```

Establecer el ámbito de las métricas en los perfiles de seguridad utilizando dimensiones

Las dimensiones son atributos que se pueden definir para obtener datos más precisos sobre las métricas y los comportamientos del perfil de seguridad. El ámbito se define proporcionando un valor o patrón que se utiliza como filtro. Por ejemplo, puede definir una dimensión de filtrado de temas que aplique una métrica solo a los temas de MQTT que coincidan con un determinado valor; por ejemplo "data/bulb/+/activity". Para obtener más información acerca de cómo definir una dimensión que pueda utilizar en el perfil de seguridad, consulte [CreateDimension](#).

Se pueden utilizar comodines de MQTT en los valores de las dimensiones. Los comodines de MQTT le permiten suscribirse a varios temas simultáneamente. Hay dos tipos diferentes de comodines: de un nivel (+) y de varios niveles (#). Por ejemplo, el valor de dimensión Data/bulb/+/activity crea una suscripción que busca correspondencias en todos los temas que existen en el mismo nivel que +. Los valores de dimensión también admiten la variable de sustitución del ID de cliente de MQTT \${iot:ClientId}.

Las dimensiones de tipo TOPIC_FILTER son compatibles con el siguiente conjunto de métricas del lado de la nube:

- Número de mensajes enviados
- Número de mensajes recibidos
- Tamaño de bytes del mensaje
- Dirección IP de origen
- Número de errores de autorización

Cómo utilizar las dimensiones en la consola

Para crear una dimensión y aplicarla a un comportamiento del perfil de seguridad

1. En el navegador [AWS IoT consola](#), en el panel de navegación expandaDefender, expandaDetectary luego seleccionePerfiles de seguridad.
2. En la página Security profiles (Perfiles de seguridad), elija Create (Crear) para agregar un nuevo perfil de seguridad o Edit (Editar) para aplicar una dimensión a un perfil de seguridad existente.
3. En la página Expected behaviors (Comportamientos esperados), seleccione una de las cinco dimensiones del lado de la nube admitidas en Metric (Métrica). Aparecerán los cuadros Dimension (Dimensión) y Dimension operator (Operador de dimensión). Para obtener información sobre las métricas del lado de la nube admitidas, consulte [Establecer el ámbito de las métricas en los perfiles de seguridad utilizando dimensiones \(p. 1026\)](#).

4. Para Dimensión, elige Añadir dimensión.
5. En la página Create a new dimension (Crear una nueva dimensión), escriba los detalles de la nueva dimensión. En Dimensions values (Valores de dimensión), se pueden utilizar los caracteres comodín de MQTT # y +, así como la variable de sustitución del ID de cliente de MQTT \${iot:ClientId}.

Create a new dimension

Dimensions control the scope of behaviors that you define in your security profiles. For example, define a dimension that monitors specific MQTT topics.

Dimension name [?](#)
Room_Temperature

Dimension type [?](#)
Topic filter

Dimension values [?](#)
/temperature/room/+

Add value

Tags

Cancel Save

6. Elija Save (Guardar).
7. Opcionalmente, puede agregar dimensiones a las métricas en Métricas adicionales que se van a conservar.
8. Para terminar de crear el comportamiento, escriba la información en los demás campos necesarios y luego seleccione Próximo.
9. Complete los pasos restantes para terminar de crear el perfil de seguridad.

Para ver las infracciones

1. En el navegador [AWS IoT consola](#), en el panel de navegación expanda **Defender**, expanda **Detect** y luego seleccione **Infracciones**.

Device Defender > Detect > Violations

Violations

Now History

Event time	Thing name	Security profile	Behavior ?	Last emitted ?
Mar 26, 2020 10:55:00 PM -0700	iotconsole-1585288160280-0	test_SP	TamperDetected	4 message(s)
Mar 26, 2020 10:55:00 PM -0700	iotconsole-1585288160280-0			Messages sent less than 1 in 5 minutes with dimension Tamper, with datapoints to Alarm: 1, and datapoints to Clear: 2

2. En el navegadorComportamiento, pausa sobre el comportamiento del que desea ver la información de infracciones.

Para ver y actualizar las dimensiones

1. En el navegador[AWS IoTconsola](#), en el panel de navegación expandaDefender, expandaDetectary luego seleccioneDimensions.
2. Seleccione la dimensión que deseé editar.
3. Seleccione Actions y luego Edit.

The screenshot shows a table titled "Dimensions (1)". The columns are "Created date", "Dimension name", "Type", and "Value". There is one row with the following values: Mar 27, 2020 3:22:51 PM -0700, Sensor_Temperature, Topic filter, /sensor/temperature/+.

Created date	Dimension name	Type	Value
Mar 27, 2020 3:22:51 PM -0700	Sensor_Temperature	Topic filter	/sensor/temperature/+

Para eliminar una dimensión

1. En el navegador[AWS IoTconsola](#), en el panel de navegación expandaDefender, expandaDetectary luego seleccioneDimensions.
2. Seleccione la dimensión que deseé eliminar.
3. Para confirmar que la dimensión no está asociada a ningún perfil de seguridad, consulte la columna Used in (Usado en) . Si la dimensión está asociada a un perfil de seguridad, abra la página Security profiles (Perfiles de seguridad) de la izquierda y edite los perfiles de seguridad a los que está asociada la dimensión. Cuando elimine la dimensión, también se eliminará el comportamiento. Si desea mantener el comportamiento, elija los puntos suspensivos y haga clic en Copy (Copiar). Continúe después con la eliminación del comportamiento. Si desea eliminar otra dimensión, siga los pasos de esta sección.

EDIT SECURITY PROFILE

Expected behaviors

STEP 1/4

Name: Temperature_Profile

Description (optional): An optional short description

Behaviors

Specify how your device should behave. You can use cloud-side metrics without a device agent deployed [learn more](#) ⓘ
Note: once created, behavior names cannot be edited. ⓘ

Name ⓘ	Metric ⓘ	Dimension (optional) ⓘ	Dimension operator ⓘ	...
Sensor_failures	Authorization failures	Sensor_Temperature	In	

Check type ⓘ	Operator ⓘ	Value	Duration ⓘ	...
Absolute value	Greater than	5	5 minutes	

Datapoints to alarm ⓘ	Datapoints to clear ⓘ
1	1

Name ⓘ	Metric ⓘ	Dimension (optional) ⓘ	Dimension operator ⓘ	...
Behavior name	Authorization failures	Select	Select	

Check type ⓘ	Operator ⓘ	Value	Duration ⓘ	...
Absolute value	Greater than	5	5 minutes	

Datapoints to alarm ⓘ	Datapoints to clear ⓘ
1	1

Add behavior

- Elija Acciones y, a continuación, elija Eliminar.

Cómo utilizar las dimensiones en el AWS CLI

Para crear una dimensión y aplicarla a un comportamiento del perfil de seguridad

- Primero cree la dimensión antes de asociarla a un perfil de seguridad. Usar [CreateDimension](#) comando para crear una dimensión:

```
aws iot create-dimension \
--name TopicFilterForAuthMessages \
--type TOPIC_FILTER \
--string-values device/+auth
```

El resultado de este comando tendrá un aspecto similar al siguiente.

```
{  
    "arn": "arn:aws:iot:us-west-2:123456789012:dimension/TopicFilterForAuthMessages",  
    "name": "TopicFilterForAuthMessages"  
}
```

2. Agregue la dimensión a un perfil de seguridad existente mediante[UpdateSecurityProfile](#) o agregue la dimensión a un nuevo perfil de seguridad mediante[CreateSecurityProfile](#). En el siguiente ejemplo, creamos un nuevo perfil de seguridad que comprueba si los mensajes a `TopicFilterForAuthMessages` tienen menos de 128 bytes y conserva el número de mensajes enviados a temas que no son de autenticación.

```
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\\\"}}]"
```

El resultado de este comando tendrá un aspecto similar al siguiente.

```
{
  "securityProfileArn": "arn:aws:iot:us-west-2:1234564789012:securityprofile/ProfileForConnectedDevice",
  "securityProfileName": "ProfileForConnectedDevice"
}
```

Para ahorrar tiempo, también puede cargar un parámetro desde un archivo en lugar de escribirlo como un valor de parámetro de línea de comandos. Para obtener más información, consulte[CargandoAWS CLI Parámetros de un archivo](#). A continuación se muestra el parámetro `behavior` en formato JSON expandido:

```
[
  {
    "criteria": {
      "comparisonOperator": "less-than",
      "consecutiveDatapointsToAlarm": 1,
      "consecutiveDatapointsToClear": 1,
      "value": {
        "count": 128
      }
    },
    "metric": "aws:message-byte-size",
    "metricDimension": {
      "dimensionName": "TopicFilterForAuthMessages"
    },
    "name": "CellularBandwidth"
  }
]
```

Para ver perfiles de seguridad con una dimensión

- Usar[ListSecurityProfiles](#)para ver perfiles de seguridad con una dimensión determinada:

```
aws iot list-security-profiles \
--dimension-name TopicFilterForAuthMessages
```

El resultado de este comando tendrá un aspecto similar al siguiente.

```
{  
    "securityProfileIdentifiers": [  
        {  
            "name": "ProfileForConnectedDevice",  
            "arn": "arn:aws:iot:us-west-2:1234564789012:securityprofile/  
ProfileForConnectedDevice"  
        }  
    ]  
}
```

Para actualizar la dimensión

- Usar[UpdateDimension](#) comando para actualizar una dimensión:

```
aws iot update-dimension \  
--name TopicFilterForAuthMessages \  
--string-values device/${iot:ClientId}/auth
```

El resultado de este comando tendrá un aspecto similar al siguiente.

```
{  
    "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"  
}
```

Para eliminar una dimensión

1. Para eliminar una dimensión, primero desconéctela de los perfils de seguridad a los que esté asociada. Usar[ListSecurityProfiles](#) para ver perfils de seguridad con una dimensión determinada.
2. Para quitar una dimensión de un perfil de seguridad, utilice el[UpdateSecurityProfile](#) comando. Introduzca toda la información que deseé conservar, pero excluya la dimensión:

```
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}}]"
```

El resultado de este comando tendrá un aspecto similar al siguiente.

```
{  
    "behaviors": [  
        {  
            "name": "ProfileForConnectedDevice",  
            "arn": "arn:aws:iot:us-west-2:1234564789012:securityprofile/  
ProfileForConnectedDevice",  
            "lastModifiedDate": 1585866222.317,  
            "stringValues": [  
                "device/${iot:ClientId}/auth"  
            ],  
            "creationDate": 1585854500.474,  
            "type": "TOPIC_FILTER",  
            "arn": "arn:aws:iot:us-west-2:1234564789012:dimension/TopicFilterForAuthMessages"  
        }  
    ]  
}
```

```
"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
}
```

- Una vez desconectada la dimensión, utilice el [DeleteDimension](#) comando para eliminar la dimensión:

```
aws iot delete-dimension \
--name TopicFilterForAuthMessages
```

Permisos

Esta sección contiene información sobre cómo configurar los roles y políticas de IAM necesarias para administrar AWS IoT Device Defender Detectar. Para obtener más información, consulte la [Guía del usuario de IAM](#).

Dar AWS IoT Device Defender Detectar permiso a para publicar alarmas en un tema de SNS

Si utiliza el `alertTargets` parámetro en [CreateSecurityProfile](#), debe especificar un rol de IAM con dos políticas: una política de permisos y una política de confianza. La política de permisos concede permiso a AWS IoT Device Defender para publicar notificaciones en su tema de SNS. La política de confianza otorga permiso a AWS IoT Device Defender para asumir el rol requerido.

Política de permisos

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

```
{  
    "Effect": "Allow",  
    "Action": [  
        "sns:Publish"  
    ],  
    "Resource": [  
        "arn:aws:sns:region:account-id:your-topic-name"  
    ]  
}  
}  
]
```

Política de confianza

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

Política para pasar roles

También necesita una política de permisos de IAM asociada al usuario de IAM que permita al usuario pasar roles. Consulte [Concesión de permisos a un usuario para transferir un rol a un servicio de 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 detección

Puede utilizar los comandos de Detect de esta sección para configurar perfiles de seguridad de Detect de ML o de Detect de reglas, para identificar y supervisar comportamientos inusuales que podrían indicar que un dispositivo está comprometido.

Comandos de las acciones de Detect de

Iniciar y administrar Ejecución de Detect

[Cancelar tarea de acciones de mitigación de detección](#)

Iniciar y administrar Ejecución de Detect
Describir la tarea de acciones de mitigación de detección
Tareas de acciones de mitigación de detección de listas
Task Iniciar detección de acciones de mitigación
Ejecuciones de acciones de mitigación de detección de listas

Comandos de las acciones

Iniciar y administrar ejecución de Dimensión
CreateDimension
DescribeDimension
ListDimensions
DeleteDimension
UpdateDimension

Comandos de las acciones de

Iniciar y administrar ejecución de CustomMetric
Crear métrica personalizada
Actualizar métrica personalizada
Describir Custom Metric
Listar métricas personalizadas
Eliminar métrica personalizada

Comandos de las acciones de

Iniciar y administrar ejecución de Security Profile
CreateSecurityProfile
AttachSecurityProfile
DeleteSecurityProfile
DescribeSecurityProfile
ListTargetsForSecurityProfile
UpdateSecurityProfile
ValidateSecurityProfileBehaviors
ListSecurityProfilesForTarget

Comandos de las acciones

Administrar alarmas y objetivos
ListActiveViolations
ListViolationEvents
DetachSecurityProfile

Comandos de las acciones de

Lista de datos de capacitación del modelo de ML
Obtener resúmenes de formación del modelo de comportamiento

Cómo utilizar AWS IoT Device Defender Detect

1. Puede usar AWS IoT Device Defender Detect solo con las métricas del lado de la nube, pero si pretende usar métricas notificadas por el dispositivo, primero debe implementar el SDK de AWS IoT en sus dispositivos conectados a AWS IoT o en las gateways de dispositivos. Para obtener más información, consulte [Envío de métricas desde dispositivos \(p. 1018\)](#).
 2. Considere la posibilidad de ver las métricas que sus dispositivos generan antes de definir comportamientos y crear alarmas. AWS IoT puede recopilar métricas de sus dispositivos, por lo que puede identificar en primer lugar un comportamiento habitual o inusual de un grupo de dispositivos o de todos los dispositivos de su cuenta. Usar [CreateSecurityProfile](#), pero especifique únicamente `losadditionalMetricsToRetainque` le interese. No especifique `behaviorsEn` este punto.
- Utilice la consola de AWS IoT para analizar sus métricas de dispositivos y determinar qué constituye un comportamiento típico para sus dispositivos.
3. Cree un conjunto de comportamientos para el perfil de seguridad. Los comportamientos contienen métricas que especifican el comportamiento normal de un grupo de dispositivos o de todos los dispositivos de su cuenta. Para obtener más información y ejemplos, consulte [Métricas del lado de la nube \(p. 1019\)](#) y [Métricas del lado del dispositivo \(p. 1003\)](#). Después de crear un conjunto de comportamientos, puede validarlos con [ValidateSecurityProfileBehaviors](#).
 4. Usar [CreateSecurityProfile](#) Acción para crear un perfil de seguridad que incluya los comportamientos. Puede utilizar el `alarmTargets` para enviar alarmas a un objetivo (un tema SNS) cuando un dispositivo vulnere un comportamiento. (Si envía alarmas usando SNS, tenga en cuenta que estas se tendrán en cuenta para la cuota de temas SNS. Si se produce una gran oleada de infracciones, podría superarse la cuota de temas de SNS. También puede utilizar las métricas de CloudWatch para comprobar las vulneraciones. Para obtener más información, consulte [Uso de las métricas de AWS IoT \(p. 433\)](#).
 5. Usar [AttachSecurityProfile](#) Acción para asociar el perfil de seguridad a un grupo de dispositivos (un grupo de objetos), a todos los objetos registrados en la cuenta, a todos los objetos no registrados o a todos los dispositivos. AWS IoT Device Defender Detect comienza comprobando si hay un comportamiento anómalo, y si se detectan vulneraciones del comportamiento, envía alarmas. Es posible que desee asociar un perfil de seguridad a todos los objetos no registrados si, por ejemplo, tiene previsto interactuar con dispositivos móviles que no están en el registro de objetos de su cuenta. Puede definir diferentes conjuntos de comportamientos para diferentes grupos de dispositivos para satisfacer sus necesidades.

Para asociar un perfil de seguridad a un grupo de dispositivos, debe especificar el ARN del grupo de objetos que los contiene. El ARN de un grupo de objetos tiene el siguiente formato:

```
arn:aws:iot:region:account-id:thinggroup/thing-group-name
```

Para asociar un perfil de seguridad a todos los objetos registrados en unCuenta de AWS(sin tener en cuenta los objetos no registrados), debe especificar un ARN con el siguiente formato:

```
arn:aws:iot:region:account-id:all/registered-things
```

Para asociar un perfil de seguridad a todos los objetos no registrados, debe especificar un ARN con el siguiente formato:

```
arn:aws:iot:region:account-id:all/unregistered-things
```

Para asociar un perfil de seguridad a todos los dispositivos, debe especificar un ARN con el siguiente formato:

```
arn:aws:iot:region:account-id:all/things
```

6. También puede realizar un seguimiento de las infracciones con el[ListActiveViolations](#), que le permite ver qué infracciones se han detectado para un perfil de seguridad o dispositivo de destino determinado.

Usar[ListViolationEvents](#)Acción para ver qué infracciones se detectaron durante un período de tiempo especificado. Puede filtrar estos resultados por un perfil de seguridad, dispositivo o estado de verificación de alarmas.

7. Puede verificar, organizar y administrar sus alarmas, marcando su estado de verificación y proporcionando una descripción de ese estado de verificación, utilizando el[Estado de verificación de Put sobre la infracción](#)action.
8. Si sus dispositivos infringen los comportamientos definidos con demasiada frecuencia o no lo suficiente, debe ajustar el comportamiento.
9. Para revisar los perfiles de seguridad que ha configurado y los dispositivos que están siendo monitorizados, utilice la[ListSecurityProfiles](#),[ListSecurityProfilesForTarget](#), y[ListTargetsForSecurityProfile](#)acciones.

Usar[DescribeSecurityProfile](#)Acción para obtener más detalles sobre un perfil de seguridad.

10. Para actualizar un perfil de seguridad, utilice la[UpdateSecurityProfile](#)action.
Usar[DetachSecurityProfile](#)Acción para desvincular un perfil de seguridad de destino de una cuenta o grupo de objetos. Usar[DeleteSecurityProfile](#)para eliminar un perfil de seguridad en su totalidad.

Acciones de mitigación

Puede usarAWS IoT Device Defenderpara emprender acciones para mitigar los problemas que se encontraron en una detección de auditoría o Detect (Detect).

Note

Las acciones de mitigación no se llevarán a cabo en los resultados de auditoría suprimidos. Para obtener más información acerca de las supresiones de búsqueda de auditorías, consulte[Supresiones de hallazgos de auditoría \(p. 974\)](#).

Acciones de mitigación de auditoría

AWS IoT Device Defender proporciona acciones predefinidas para las distintas comprobaciones de auditoría. Configura esas acciones para tu cuenta de AWS, a continuación, aplicarlas a un conjunto de resultados. Estos resultados pueden ser:

- Todos los resultados de una auditoría. Esta opción está disponible en la consola de AWS IoT y a través de la AWS CLI.
- Una lista de resultados individuales. Esta opción solo está disponible a través de la AWS CLI
- Un conjunto filtrado de los resultados de una auditoría.

En la siguiente tabla se muestran los tipos de las comprobaciones de auditoría y las acciones de mitigación compatibles para cada una de ellas:

Comprobación de auditoría y acción de mitigación

Comprobación de auditoría	Acciones de mitigación admitidas
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 las comprobaciones de auditoría admiten la publicación de los resultados de la auditoría en Amazon SNS para que pueda tomar acciones personalizadas en respuesta a la notificación. Cada tipo de comprobación de auditoría puede admitir acciones de mitigación adicionales:

REVOKE_CA_CERT_CHECK

- Cambiar el estado del certificado para marcarlo como inactivo en AWS IoT.

DEVICE_CERTIFICATE_SHARED_CHECK

- Cambiar el estado del certificado del dispositivo para marcarlo como inactivo en AWS IoT.
- Añadir dispositivos que utilizan dicho certificado a un grupo de objetos.

UNAUTHENTICATED_COGNITO_ROLE_OVERLY_PERMISSIVE_CHECK

- No se admiten otras acciones.

AUTHENTICATED_COGNITO_ROLE_OVERLY_PERMISSIVE_CHECK

- No se admiten otras acciones.

IOT_POLICY_OVERLY_PERMISSIVE_CHECK

- Añadir una versión de política de AWS IoT en blanco para restringir los permisos.

CA_CERT_APPROACHING_EXPIRATION_CHECK

- Cambiar el estado del certificado para marcarlo como inactivo en AWS IoT.

CONFLICTING_CLIENT_IDS_CHECK

- No se admiten otras acciones.

DEVICE_CERT_APPROACHING_EXPIRATION_CHECK

- Cambiar el estado del certificado del dispositivo para marcarlo como inactivo en AWS IoT.
- Añadir dispositivos que utilizan dicho certificado a un grupo de objetos.

DEVICE_CERTIFICATE_KEY_QUALITY_CHECK

- Cambiar el estado del certificado del dispositivo para marcarlo como inactivo en AWS IoT.
- Añadir dispositivos que utilizan dicho certificado a un grupo de objetos.

CA_CERTIFICATE_KEY_QUALITY_CHECK

- Cambiar el estado del certificado para marcarlo como inactivo en AWS IoT.

REVOKE_DEVICE_CERT_CHECK

- Cambiar el estado del certificado del dispositivo para marcarlo como inactivo en AWS IoT.
- Añadir dispositivos que utilizan dicho certificado a un grupo de objetos.

LOGGING_DISABLED_CHECK

- Habilitar el registro.

AWS IoT Device Defender admite los siguientes tipos de acciones de mitigación en los resultados de auditoría:

Tipo de acción	Notas
ADD_THINGS_TO_THING_GROUP	Debe especificar el grupo a la que desea añadir los dispositivos. También debe especificar si la suscripción a uno o varios grupos dinámicos debe anularse si se supera el número máximo de grupos a los que el objeto puede pertenecer.
ENABLE_IOT_LOGGING	Debe especificar el nivel de registro y el rol con permisos para el registro. No puede especificar un nivel de registro DISABLED.

Tipo de acción	Notas
PUBLISH_FINDING_TO_SNS	Debe especificar el tema en el que se debe publicar la búsqueda.
REPLACE_DEFAULT_POLICY_VERSION	Debe especificar el nombre de la plantilla. Sustituye la versión de la política predeterminada con una política en blanco. En este momento solo es compatible un valor de BLANK_POLICY.
UPDATE_CA_CERTIFICATE	Debe especificar el nuevo estado para el certificado de CA. En este momento solo es compatible un valor de DEACTIVATE.
UPDATE_DEVICE_CERTIFICATE	Debe especificar el nuevo estado para el certificado del dispositivo. En este momento solo es compatible un valor de DEACTIVATE.

Al configurar acciones estándar para los problemas encontrados durante una auditoría, puede responder a estos problemas de forma coherente. Utilizar estas acciones de mitigación definidas también le ayuda a resolver los problemas con mayor rapidez y con menos posibilidad de que se produzcan errores.

Important

La aplicación de acciones de mitigación que cambian los certificados, añaden objetos a un nuevo grupo de objetos o sustituyen la política puede tener un impacto en sus dispositivos y aplicaciones. Por ejemplo, es posible que los dispositivos no puedan conectarse. Tenga en cuenta las implicaciones de las acciones de mitigación antes de aplicarlas. Es posible que tenga que adoptar otras medidas para corregir problemas antes de que los dispositivos y las aplicaciones puedan funcionar con normalidad. Por ejemplo, es posible que tenga que proporcionar certificados de dispositivo actualizados. Las acciones de mitigación pueden ayudarle a limitar rápidamente los riesgos pero, aún así, tiene que tomar medidas correctoras para resolver los problemas.

Algunas acciones, como, por ejemplo, reactivar un certificado de dispositivo, solo se pueden realizar manualmente. AWS IoT Device Defender no proporciona un mecanismo para revertir automáticamente las acciones de mitigación que se han aplicado.

Detección de acciones de mitigación

AWS IoT Device Defender admite los siguientes tipos de acciones de mitigación en Detect alarms:

Tipo de acción	Notas
ADD_THINGS_TO_THING_GROUP	Debe especificar el grupo a la que desea añadir los dispositivos. También debe especificar si la suscripción a uno o varios grupos dinámicos debe anularse si se supera el número máximo de grupos a los que el objeto puede pertenecer.

Cómo definir y administrar las acciones de mitigación

Puede utilizar la AWS IoT consola o AWS CLI para definir y administrar las acciones de mitigación para su cuenta de AWS.

Creación de acciones de mitigación

Cada acción de mitigación que defina es una combinación de un tipo de acción predefinida y los parámetros específicos de su cuenta.

Para utilizar la consola de AWS IoT para crear acciones de mitigación

1. Abra la [consola de AWS IoT](#).
2. En el panel de navegación de la izquierda, seleccione Defender y, a continuación, elija Mitigation Actions (Acciones de mitigación).
3. En la página Mitigation Actions (Acciones de mitigación), elija Create (Crear).

Create a new mitigation action

You can use AWS IoT Device Defender to take actions to mitigate issues that were found during an audit. AWS IoT Device Defender provides predefined actions for the different audit checks. You can configure those actions for your AWS account and then apply them to a set of findings. [Learn more](#)

Action name [?](#)

Action type [?](#)

Permissions

Please create or select a role with the following mitigation action type specific permission(s) and trust relationship.

Required permissions: [Manage your service permissions](#)

- ▶ Permissions
- ▶ Trust relationships

You can also attach an action specific managed policy to an existing role, or create a new role with the required managed policy attached.

Action execution role [?](#)

<input type="text" value="IoTMitigationActionErrorLoggingRole"/> Managed policy attached ✓	Create Role	Select
---	-----------------------------	------------------------

Parameters

Role for logging [?](#)
 [Clear](#) [Select](#)

Log level [?](#)

Tags

Tag name Value [Clear](#)

[Add another](#)

[Cancel](#) [Save](#)

4. En la página Create a Mitigation Action (Crear una acción de mitigación), en Action name (Nombre de la acción), escriba un nombre único para la acción de mitigación.
5. En Action type (Tipo de acción), especifique el tipo de acción que desea definir.
6. Cada tipo de acción solicita un conjunto diferente de parámetros. Escriba los parámetros de la acción. Por ejemplo, si elige el tipo de acción Añadir objetos al grupo de objetos, debe elegir el grupo de destino y seleccionar o quitar la marca de verificación de Override dynamic groups (Anular grupos dinámicos).

7. En Action execution role (Rol de ejecución de la acción), elija el rol bajo cuyos permisos se aplica la acción.
 8. Elija Guardar para guardar la acción de mitigación en su cuenta de AWS.

Para utilizar la AWS CLI para crear acciones de mitigación

- Usar [CreateMitigationAction](#) para crear la acción de mitigación. El nombre único que dé a la acción se utiliza cuando se aplica dicha acción a los resultados de la auditoría. Elija un nombre significativo.

Para utilizar la consola de AWS IoT para ver y modificar las acciones de mitigación

1. Abra la [consola de AWS IoT](#).
 2. En el panel de navegación de la izquierda, seleccione Defender y, a continuación, elija Mitigation Actions (Acciones de mitigación).

La sección **Acciones de mitigación** muestra una lista de todas las acciones de mitigación definidas para su cuenta de AWS.

Device Defender > Mitigation actions

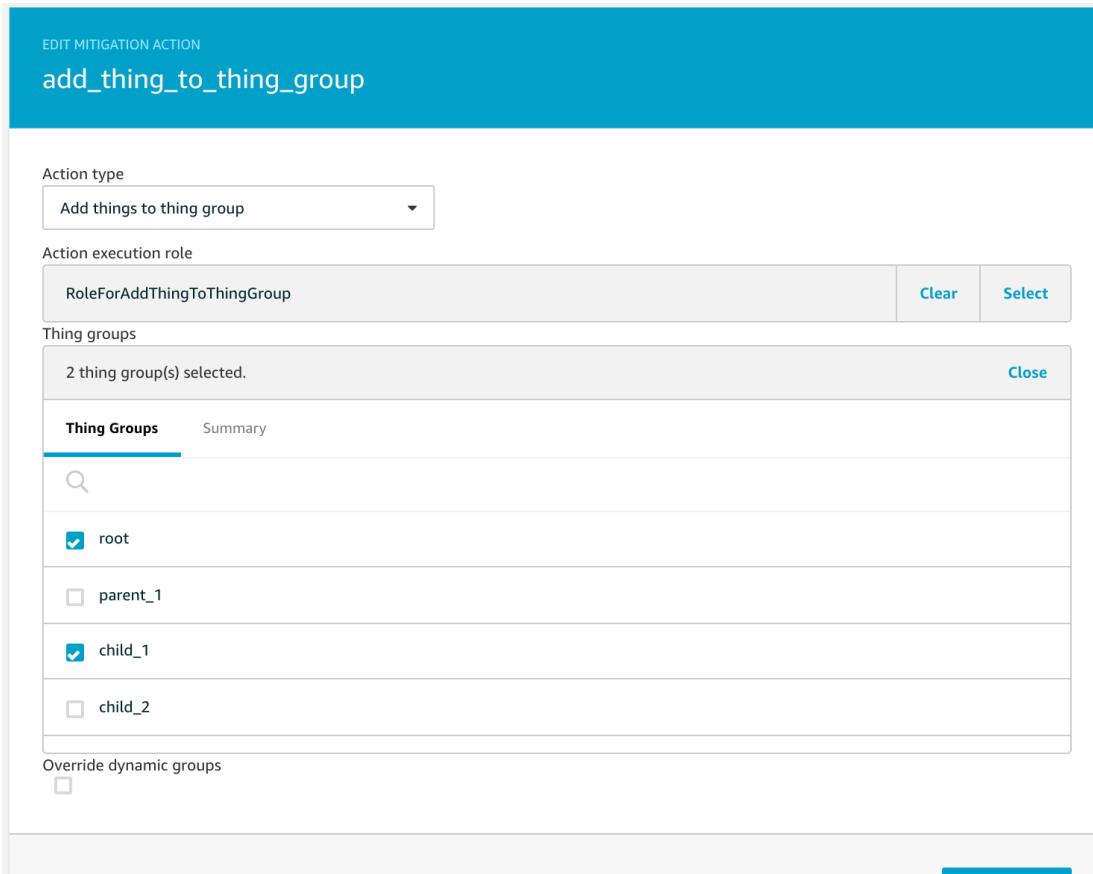
Mitigation actions (4)

Create  

1-4 of 4

Created date	Action name	ARN	...
Jun 10, 2019 10:09:53 AM -0700	enable_logging	arn:aws:iot:us-east-1:  :mitigationa... 	
Jun 6, 2019 6:08:47 PM -0700	sns_publish	arn:aws:iot:us-east-1:  :mitigationa... 	
Jun 6, 2019 6:08:26 PM -0700	replace_default_policy_version	arn:aws:iot:us-east-1:  :mitigationa... 	
Jun 3, 2019 10:51:16 PM -0700	add_thing_to_thing_group	arn:aws:iot:us-east-1:  :mitigationa... 	

3. Elija el enlace del nombre de la acción de la acción de mitigación que desea cambiar.
 4. Realice los cambios que desee en la acción de mitigación. Como el nombre de la acción de mitigación se utiliza para identificarla, no puede cambiar el nombre.



- Elegir Guardar para guardar los cambios en la acción de mitigación en su cuenta de AWS.

Para utilizar la AWS CLI para obtener una lista de acciones de mitigación

- Usar [Acción de mitigación de listas](#) para mostrar las acciones de mitigación. Si desea cambiar o eliminar una acción de mitigación, anote del nombre.

Para utilizar la AWS CLI para actualizar una acción de mitigación

- Usar [UpdateMitigationAction](#) para cambiar la acción de mitigación.

Para utilizar la consola de AWS IoT para eliminar una acción de mitigación

- Abra la [consola de AWS IoT](#).
- En el panel de navegación de la izquierda, seleccione Defender y, a continuación, elija Mitigation Actions (Acciones de mitigación).

La [Acciones de mitigación](#) muestra todas las acciones de mitigación definidas para su cuenta de AWS.

- Elija los puntos suspensivos (...) para la acción de mitigación que desea eliminar y, a continuación, haga clic en Eliminar.

Para utilizar la AWS CLI para eliminar acciones de mitigación

- Usar [UpdateMitigationAction](#) para cambiar la acción de mitigación.

Para utilizar la consola de AWS IoT para ver los detalles de una acción de mitigación

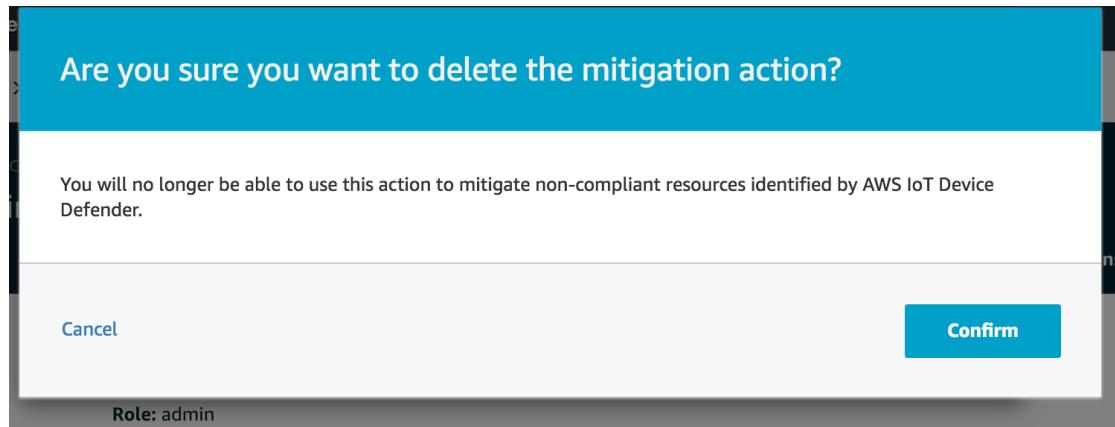
1. Abra la [consola de AWS IoT](#).
2. En el panel de navegación de la izquierda, seleccione Defender y, a continuación, elija Mitigation Actions (Acciones de mitigación).

The screenshot shows the 'Mitigation actions' section of the AWS IoT Device Defender console. At the top, there's a breadcrumb navigation: 'Device Defender > Mitigation actions'. On the right side, there are three buttons: 'Create' (blue), 'Bell icon' (blue), and a circular arrow icon. Below the header, it says 'Mitigation actions (4)'. A table lists four mitigation actions:

Created date	Action name	ARN	More
Jun 10, 2019 10:09:53 AM -0700	enable_logging	arn:aws:iot:us-east-1:...:mitigationa...	...
Jun 6, 2019 6:08:47 PM -0700	sns_publish	arn:aws:iot:us-east-1:...:mitigationa...	...
Jun 6, 2019 6:08:26 PM -0700	replace_default_policy_version	arn:aws:iot:us-east-1:...:mitigationa...	...
Jun 3, 2019 10:51:16 PM -0700	add_thing_to_thing_group	arn:aws:iot:us-east-1:...:mitigationa...	...

La Acción de mitigación muestra todas las acciones de mitigación definidas para su cuenta de AWS.

3. Elija el enlace del nombre de la acción de la acción de mitigación que desea cambiar.
4. En la ventana Are you sure you want to delete the mitigation action (¿Seguro que desea eliminar la acción de mitigación?), elija Confirm (Confirmar).



Para utilizar la AWS CLI para ver los detalles de una acción de mitigación

- Usar [DescribeMitigationAction](#) para ver los detalles de la acción de mitigación.

Aplicación de acciones de mitigación

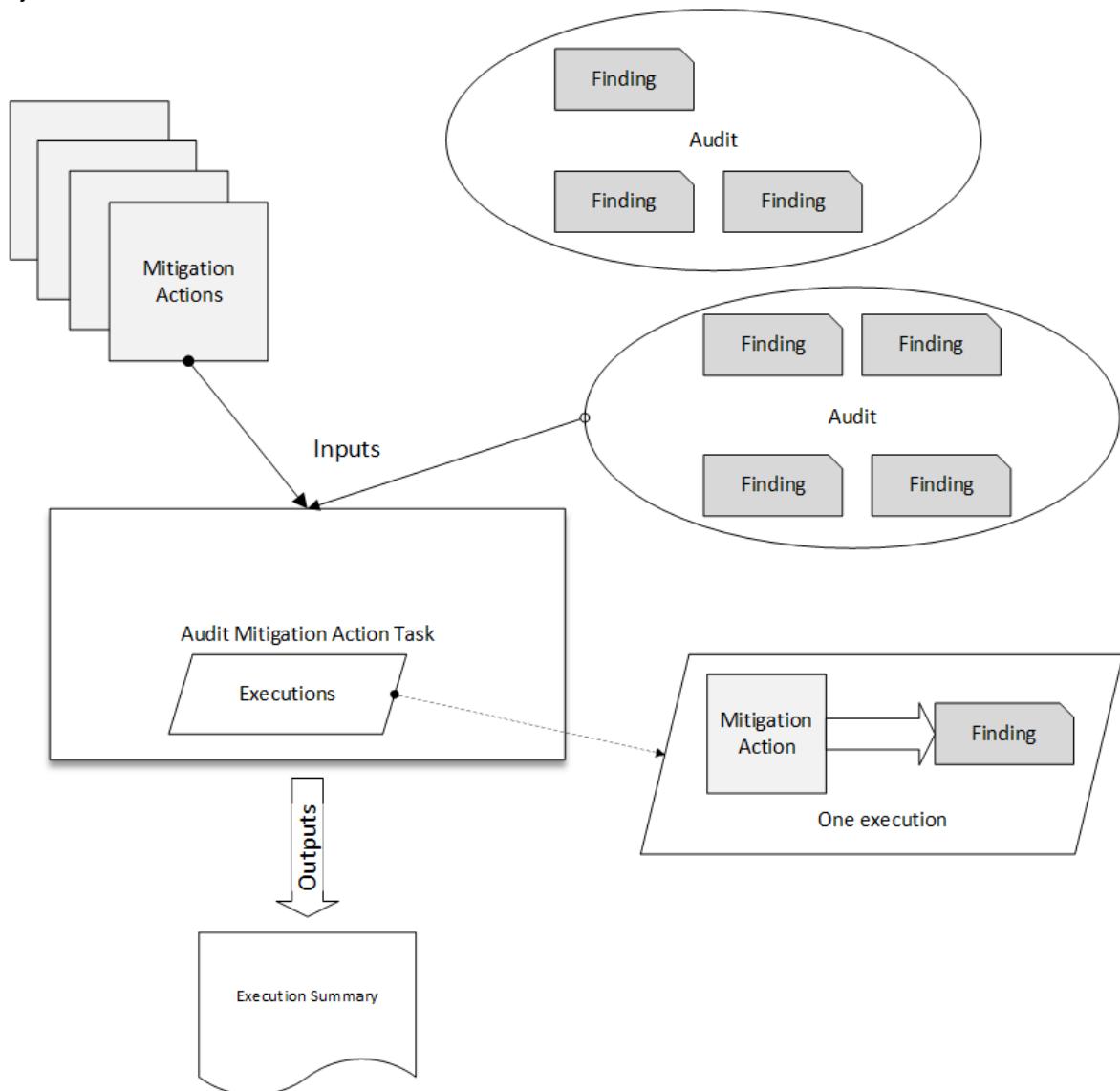
Después de haber definido un conjunto de acciones de mitigación, puede aplicar esas acciones a los resultados de una auditoría. Cuando se aplican acciones, empieza una tarea de acciones de mitigación de auditoría. Esta tarea podría tardar un tiempo en completarse, en función del conjunto de resultados y las acciones que se aplican a ellos. Por ejemplo, si tiene un gran grupo de dispositivos cuyos certificados han caducado, puede tardar algún tiempo en desactivar todos estos certificados o en mover esos dispositivos a un grupo de cuarentena. Otras acciones, como la habilitación del registro, se pueden realizar con rapidez.

Puede ver la lista de ejecuciones de acciones y cancelar una ejecución que aún no se ha completado. Las acciones ya realizadas como parte de la ejecución de la acción cancelada no se revierten. Si está

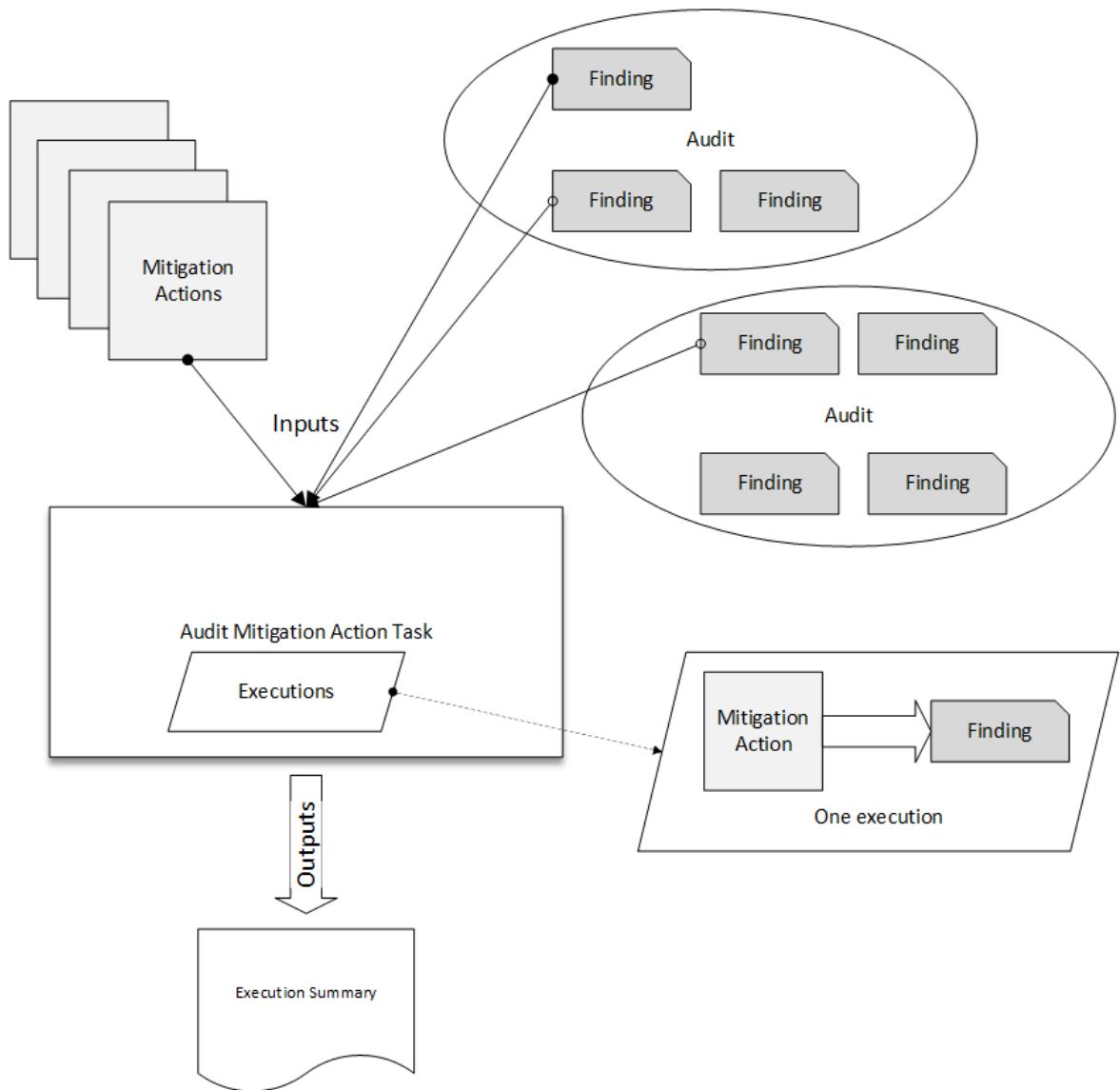
aplicando varias acciones a un conjunto de resultados y una de esas acciones produce un error, se omiten las acciones posteriores para ese resultado (pero se siguen aplicando a otros resultados). El estado de la tarea para el resultado es FAILED. El taskStatus se establece como error (Failed) si una o varias de las acciones fracasan cuando se aplican a los resultados. Las acciones se aplican en el orden en que estén especificadas.

Cada ejecución de acción aplica a un conjunto de acciones a un destino. Ese destino puede ser una lista de resultados o puede ser todos los resultados de una auditoría.

En el siguiente diagrama se muestra cómo puede definir una tarea de mitigación de auditoría para que tome todos los resultados de una auditoría y les aplique un conjunto de acciones. Una única ejecución aplica una acción a un resultado. La tarea de acciones de mitigación de auditoría genera un resumen de ejecución.



En el siguiente diagrama se muestra cómo puede definir una tarea de mitigación de auditoría para que tome una lista de resultados individuales de una o varias auditorías y les aplique un conjunto de acciones. Una única ejecución aplica una acción a un resultado. La tarea de acciones de mitigación de auditoría genera un resumen de ejecución.



Puede utilizar la AWS CLI o la consola de AWS IoT para aplicar acciones de mitigación.

Para usar la consola AWS IoT para aplicar acciones de mitigación iniciando la ejecución de una acción

1. Abra la [consola de AWS IoT](#).
2. En el panel de navegación de la izquierda, seleccione Defender, elija Auditoría y, a continuación, elija Resultados.

Check name	Severity	Non-compliant	% Resources	Mitigation
CA certificate revoked but device certificates still a...	Critical	79	83.2%	Review & deactivate (i)
CA certificate expiring	Medium	79	83.2%	Reprovision & deactivate (i)
Device certificate expiring	Medium	50	92.6%	Reprovision & deactivate (i)
Revoked device certificate still active	Medium	50	92.6%	Reprovision & revoke (i)
Logging disabled	Low	1	100%	Enable logging (i)

3. Elija el nombre de la auditoría a la que desea aplicar acciones.

Starting mitigation actions for the current audit report will start execution of the given actions against all findings in the report. If certain checks are not displayed in the list, it is because no actions have been configured with the type that is applicable for the check. [Create mitigation action](#)

Task name [\(i\)](#)
8115481b332374e3309c13050320323c

Select options for Device certificate expiring [\(i\)](#)

Select actions [\(i\)](#)
1 actions selected [Expand](#)

Select reason codes [\(i\)](#)
2 reason codes selected [Expand](#)

Cancel [Confirm](#)

4. Selec Start Mitigation Actions (Iniciar acciones de mitigación). Este botón no está disponible si todos las comprobaciones son correctas.
5. En Are you sure that you want to start mitigation action task (¿Seguro que desea iniciar la tarea de acción de mitigación?), el nombre de la tarea toma de forma predeterminada el ID de la auditoría, pero puede cambiarlo a algo más significativo.
6. Para cada tipo de comprobación que tenga uno o varios resultados no conformes en la auditoría, puede elegir una o más acciones para aplicar. Solo se muestran las acciones que son válidas para el tipo de comprobación.

Note

Si no ha configurado las acciones para su cuenta de AWS, la lista de acciones está vacía. Puede elegir el enlace click here (haga clic aquí) para crear una o varias acciones de mitigación.

7. Cuando haya especificado todas las acciones que se van a aplicar, seleccione Confirm (Confirmar).

Para usar la AWS CLI para aplicar acciones de mitigación iniciando la ejecución de una acción de mitigación de auditoría

1. Si desea aplicar acciones a todos los resultados de la auditoría, utilice el [ListAuditTasks](#) para encontrar el ID de la tarea.
2. Si desea aplicar acciones solo a resultados seleccionados, utilice el [ListAuditFindings](#) para obtener los ID de búsqueda.
3. Usar [ListMitigationActions](#) y anote los nombres de las acciones de mitigación que se van a aplicar.
4. Usar [StartAuditMitigationActionsTask](#) para aplicar acciones al destino. Anote el ID de la tarea. Puede utilizar el ID para comprobar el estado de la ejecución de la acción, revisar los detalles o cancelarla.

Para utilizar la consola de AWS IoT para ver sus ejecuciones de acciones

1. Abra la [consola de AWS IoT](#).
2. En el panel de navegación de la izquierda, seleccione Defender y, a continuación, elija Action Executions (Ejecuciones de acciones).

The screenshot shows a table titled "Action tasks (1)". The table has three columns: "Date", "Name", and "Status". There is one row with the following data:

Date	Name	Status
Jun 6, 2019 6:09:07 PM -0700	ff82164a6439e6024e83b4fc104817d7	Completed

Una lista de tareas de acción muestra cuando cada se inició una y su estado actual.

3. Elija el enlace Name (Nombre) para ver los detalles de la tarea. Los detalles incluyen todas las acciones que aplica la tarea, su destino y su estado.

The screenshot shows a detailed view of a mitigation action execution task. At the top, it displays the task ID: ff82164a6439e6024e83b4fc104817d7. Below this, there are sections for "Details" and "Check summary".

Details

Status	COMPLETED
Started at	Jun 6, 2019 6:09:07 PM -0700
Completed at	Jun 6, 2019 6:09:09 PM -0700

Check summary

Check name	Failed	Successful	Skipped	Canceled	Total	Executions
IoT policies overly permissive	0	2	0	0	2	Show

Puede utilizar los filtros Show executions for (Mostrar ejecuciones para) en tipo de acciones o estados de acción.

4. Para ver los detalles de la tarea, en Executions (Ejecuciones), elija Show (Mostrar).

The screenshot shows the AWS Device Defender Audit Action executions page. At the top, it displays the path: Device Defender > Audit > Action executions > ff82164a6439e6024e83b4fc104817d7 >. Below this, a dark header bar contains the text "MITIGATION ACTION EXECUTION TASK" and the ID "ff82164a6439e6024e83b4fc104817d7". The main content area has a heading "IoT policies overly permissive". Underneath, there's a section titled "Action executions (4)" with a "Show executions for" dropdown set to "All actions" and "All status". A table follows, showing 1-4 of 4 entries:

Started at	Status	Action	Finding
Jun 6, 2019 6:09:08 PM -0700	Completed	sns_publish	053cff17-1da4-4479-996b-8b...
Jun 6, 2019 6:09:08 PM -0700	Completed	replace_default_policy_version	053cff17-1da4-4479-996b-8b...
Jun 6, 2019 6:09:08 PM -0700	Completed	replace_default_policy_version	2b966f76-b499-4986-836c-f8...

Para utilizar la AWS CLI para mostrar una lista de tareas iniciadas

1. Usar [ListAuditMitigationActionsTasks](#) para ver las tareas de acciones de mitigación de auditoría. Puede proporcionar filtros para limitar los resultados. Si desea ver los detalles de la tarea, anote el ID de la tarea.
2. Usar [ListAuditMitigationActionsExecutions](#) para ver los detalles de ejecución de una tarea de acciones de mitigación de auditoría en particular.
3. Usar [DescribeAuditMitigationActionsTask](#) para ver los detalles de la tarea, como los parámetros especificados cuando se inició la tarea.

Para utilizar la AWS CLI para cancelar una tarea de acciones de mitigación de auditoría en ejecución

1. Usar [ListAuditMitigationActionsTasks](#) para encontrar el ID de la tarea cuya ejecución desea cancelar. Puede proporcionar filtros para limitar los resultados.
2. Usar [Ejecuciones de acciones de mitigación de detección de listas](#), utilizando el ID de la tarea, para cancelar la tarea de acciones de mitigación de auditoría. No es posible cancelar tareas que se han completado. Cuando se cancela una tarea, las acciones restantes no se aplican, pero las acciones de mitigación que ya se han aplicado no se revierten.

Permisos

Para cada acción de mitigación que defina, debe proporcionar el rol utilizado para aplicar dicha acción.

Permisos para acciones de mitigación

Tipo de acción	Plantilla de política de permisos
UPDATE_DEVICE_CERTIFICATE	{

Tipo de acción	Plantilla de política de permisos
	<pre>"Version":"2012-10-17", "Statement":[{ "Effect":"Allow", "Action":["iot:UpdateCertificate"], "Resource":["*"] } }</pre>
UPDATE_CA_CERTIFICATE	<pre>{ "Version":"2012-10-17", "Statement":[{ "Effect":"Allow", "Action":["iot:UpdateCACertificate"], "Resource":["*"] }] }</pre>
ADD_THINGS_TO_THING_GROUP	<pre>{ "Version":"2012-10-17", "Statement":[{ "Effect":"Allow", "Action":["iot>ListPrincipalThings", "iot:AddThingToThingGroup"], "Resource":["*"] }] }</pre>

Tipo de acción	Plantilla de política de permisos
REPLACE_DEFAULT_POLICY_VERSION	<pre>{ "Version": "2012-10-17", "Statement": [{ "Effect": "Allow", "Action": ["iot:CreatePolicyVersion"], "Resource": ["*"] }] }</pre>
ENABLE_IOT_LOGGING	<pre>{ "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>"] }] }</pre>

Tipo de acción	Plantilla de política de permisos
PUBLISH_FINDING_TO_SNS	<pre>{ "Version": "2012-10-17", "Statement": [{ "Effect": "Allow", "Action": ["sns:Publish"], "Resource": ["<The SNS topic to which the finding is published>"] }] }</pre>

Para todos los tipos de acciones de mitigación, utilice la siguiente plantilla de política de confianza:

```
{
    "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 las acciones de mitigación

Puede utilizar estos comandos de acciones de mitigación para definir un conjunto de acciones para su cuenta de AWS que más tarde se pueden aplicar a uno o varios conjuntos de resultados de auditoría. Existen tres categorías de comandos:

- Los que se usan para definir y administrar acciones.
- Los que se usan para iniciar y administrar la aplicación de esas acciones a los resultados de auditoría.
- Los que se usan para iniciar y administrar la aplicación de esas acciones a Detectar alarmas.

Comandos de las acciones de mitigación

Definir y administrar acciones	Iniciar y administrar ejecución de auditoría	Iniciar y administrar ejecución de Detectar
CreateMitigationAction	CancelAuditMitigationActionsTask	Cancelar tarea de acciones de mitigación de detección
DeleteMitigationAction	DescribeAuditMitigationActionsTask	Describir la tarea de acciones de mitigación de detección
DescribeMitigationAction	ListAuditMitigationActionsTasks	Tareas de acciones de mitigación de detección de listas
ListMitigationActions	StartAuditMitigationActionsTask	Task Iniciar detección de acciones de mitigación
UpdateMitigationAction	ListAuditMitigationActionsExecutions	Ejecuciones de acciones de mitigación de detección de listas

Uso de AWS IoT Device Defender con otros servicios de AWS.

Uso de AWS IoT Device Defender con dispositivos en ejecución AWS IoT Greengrass

AWS IoT Greengrass proporciona integración preconstruida con AWS IoT Device Defender para supervisar los comportamientos de los dispositivos de forma continua.

- [Integrar Device Defender de AWS IoT Greengrass V1](#)
- [Integrar Device Defender de AWS IoT Greengrass V2](#)

Uso de AWS IoT Device Defender con FreeRTOS y dispositivos integrados

Para utilizar AWS IoT Device Defender en un dispositivo FreeRTOS, el dispositivo debe tener el [SDK para Embedded C de FreeRTOS](#) o la [AWS Biblioteca IoT Device Defender de](#) instalado. El SDK C integrado de FreeRTOS incluye la [AWS Biblioteca IoT Device Defender](#). Para obtener información acerca de cómo efectuar la integración AWS IoT Device Defender con sus dispositivos FreeRTOS, consulte las siguientes demostraciones:

- [AWS IoT Device Defender para métricas estándar FreeRTOS y demostraciones de métricas personalizadas](#)
- [Uso del agente MQTT para enviar métricas a AWS IoT Device Defender](#)
- [Uso de la biblioteca principal de MQTT para enviar métricas a AWS IoT Device Defender](#)

Para utilizar AWS IoT Device Defender en un dispositivo integrado sin FreeRTOS, el dispositivo debe tener el [AWS SDK para IoT para Embedded](#) o la [AWS Biblioteca IoT Device Defender de](#). La [AWS IoT Device Defender](#) para

IoT Embedded C incluye AWS Biblioteca IoT Device Defender. Para obtener información acerca de cómo efectuar la integración AWS IoT Device Defender con tus dispositivos integrados, consulta las siguientes demostraciones, [AWS IoT Device Defender para AWS Demostraciones de métricas personalizadas y estándar del SDK integrado de IoT](#).

Uso de AWS IoT Device Defender con AWS IoT Device Management

Puede usar AWS IoT Device Management indexación de flotas para indexar, buscar y agregar su AWS IoT Device Defender Detecte infracciones. Después de que los datos de infracciones de Device Defender estén indexados en la indexación de flotas, puede acceder y consultar los datos de infracciones de Device Defender desde aplicaciones de Fleet Hub, crear alarmas de flotas basadas en datos de infracciones para supervisar anomalías en la flota de dispositivos y ver las alarmas de flotas en los paneles de Fleet Hub.

Note

Función de indexación de flotas para admitir la indexación AWS IoT Device Defender Los datos de infracciones se encuentran en versión preliminar para AWS IoT Device Management y está sujeta a cambios.

- Administración del índice de flotas
- Sintaxis de la consulta
- Gestión de la indexación de flotas para aplicaciones de Fleet
- Introducción

Prevención del suplente confuso entre servicios

El problema del suplente confuso es un problema de seguridad en el que una entidad que no tiene permiso para realizar una acción puede obligar a una entidad con más privilegios a realizar la acción. En AWS, la suplantación entre servicios puede dar lugar al problema del suplente confuso. La suplantación entre servicios puede producirse cuando un servicio (el servicio que lleva a cabo las llamadas) llama a otro servicio (el servicio al que se llama). El servicio que lleva a cabo las llamadas se puede manipular para utilizar sus permisos a fin de actuar en función de los recursos de otro cliente a través del servicio que lleva a cabo las llamadas de una manera en la que no debe tener permiso para acceder. Para evitarlo, AWS proporciona herramientas que lo ayudan a proteger sus datos para todos los servicios con entidades principales de servicio a las que se les ha dado acceso a los recursos de su cuenta.

Hay tres recursos AWS IoT Device Defender accesos de usted que pueden verse afectados por el confuso problema de seguridad de la profundidad, la ejecución de auditorías, el envío de notificaciones SNS por infracciones del perfil de seguridad y la ejecución de acciones de mitigación. Para cada una de estas acciones, los valores de `aws:SourceArn` debe ser la siguiente:

- Para los recursos transferidos `UpdateAccountAuditConfiguration` API (atributos `RoleArn` y `NotificationTargetRoleArn`), debe reducir el alcance de la política de recursos mediante `aws:SourceArn` como `arn:partition:iot:region:accountId:.`
- Para los recursos transferidos `CreateMitigationAction` API (el atributo `RoleArn`) que debe reducir el alcance de la política de recursos mediante `aws:SourceArn` como `arn:partition:iot:region:accountId:migrationaction/migrationaction/.`
- Para los recursos transferidos `CreateSecurityProfile` API (el atributo `alertTargets`), debe reducir el alcance de la política de recursos mediante `aws:SourceArn` como `arn:partition:iot:region:accountId:securityprofile/securityprofile/.`

La forma más eficaz de protegerse contra el confuso problema del diputado es utilizar `aws:SourceArn` clave de contexto de condición global con el ARN completo del recurso. Si no conoce

el ARN completo del recurso o si está especificando varios recursos, utilice elaws:SourceArnclave de condición de contexto global con comodines (*) para las partes desconocidas del ARN. Por ejemplo, arn:aws:servicename:*:123456789012:*.

En el siguiente ejemplo se muestra cómo puede utilizar elaws:SourceArnaws:SourceAccountclaves de contexto de condición globales de AWS IoT Device Defender para evitar el problema del suplente confuso.

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

Prácticas recomendadas de seguridad para agentes de dispositivos

Privilegios mínimos

Al proceso del agente se debe otorgar solo los permisos mínimos necesarios para realizar sus tareas.

Mecanismos básicos

- El agente debe ejecutarse como un usuario no raíz.
- El agente se debe ejecutar como un usuario dedicado, en su propio grupo.
- Al usuario/grupo se le deben otorgar permisos de solo lectura sobre los recursos necesarios para recopilar y transmitir métricas.
- Ejemplo: solo lectura en /proc /sys para el agente de muestra.
- Para obtener un ejemplo de cómo configurar un proceso para que se ejecute con permisos reducidos, consulte las instrucciones de configuración que se incluyen con el agente de muestra de Python.

Hay una serie de mecanismos Linux conocidos que pueden ayudarle a restringir o aislar aún más el proceso de su agente:

Mecanismos avanzados

- [CGroups](#)
- [SELinux](#)
- [Chroot](#)

- [Espacios de nombres Linux](#)

Resiliencia operativa

Un proceso de agente debe ser resiliente a errores y excepciones operativas inesperados y no debe bloquearse ni salir de forma permanente. El código debe manejar con soltura excepciones y, como precaución, debe configurarse para reiniciarse automáticamente en caso de terminación inesperada (por ejemplo, debido a reinicios del sistema o excepciones no detectadas).

Dependencias mínimas

Un agente debe usar el menor número posible de dependencias (es decir, bibliotecas de terceros) en su implementación. Si el uso de una biblioteca se justifica debido a la complejidad de una tarea (por ejemplo, Transport Layer Security), use solo dependencias bien mantenidas y establezca un mecanismo para mantenerlas actualizadas. Si las dependencias agregadas contienen funcionalidades que el agente no usa y que están activas de manera predeterminada (por ejemplo, abrir puertos, sockets de dominio), desactívelas en su código o por medio de los archivos de configuración de la biblioteca.

Aislamiento de procesos

Un proceso de agente solo debe contener la funcionalidad requerida para realizar la recopilación y la transmisión de métricas del dispositivo. No debe aprovecharse de otros procesos del sistema como un contenedor ni implementar funcionalidad para otros casos de uso fuera de su ámbito. Además, el proceso del agente debe abstenerse de crear canales de comunicación de entrada como puertos de socket de dominio y de servicio de red que permitirían que procesos locales o remotos interfiriesen en su funcionamiento e influyesen en su integridad y aislamiento.

Sigilo

El nombre de un proceso de agente no debe contener palabras clave como seguridad, monitorización o auditoría que indiquen su finalidad y valor de seguridad. Se debe optar por nombres genéricos de códigos y nombres de procesos aleatorios y únicos en cada dispositivo. Se debe seguir el mismo principio al asignar nombre al directorio en el que residen los binarios del agente y los nombres y valores de los argumentos del proceso.

Información mínima compartida

Ningún artefacto de agente implementado en los dispositivos debe contener información confidencial como credenciales con privilegios, depuración ni código muerto, o comentarios insertados o archivos de documentación que revelen detalles sobre el procesamiento del lado del servidor de métricas recopiladas por el agente u otros detalles sobre sistemas backend.

Transport Layer Security

Para establecer canales seguros de TLS para la transmisión de datos, un agente debe hacer cumplir todas las validaciones del lado del cliente, como la validación de certificados y nombres de dominio en el nivel de la aplicación, si no están habilitadas de manera predeterminada. Además, un agente debe usar un almacén de certificados raíz que contenga entidades de confianza y no contenga certificados que pertenezcan a emisores de certificados atacados.

Implementación segura

Cualquier mecanismo de implementación del agente, como inserción o sincronización de código y repositorios que contengan sus binarios, código fuente y cualquier archivo de configuración (incluidos certificados raíz de confianza), debe controlarse para evitar la inserción o alteración no autorizada del código. Si el mecanismo de implementación se basa en la comunicación de red, se deben usar métodos criptográficos para proteger la integridad de los artefactos de implementación en tránsito.

Documentación adicional

- [Seguridad en AWS IoT \(p. 293\)](#)
- [Descripción del AWS IoT Modelo de seguridad](#)
- [Redhat: A Bite of 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](#)

Asesor de dispositivos

Device Advisors es una capacidad de prueba totalmente administrada y basada en la nube para validar dispositivos IoT durante el desarrollo de software de dispositivos. Device Advisor proporciona pruebas predefinidas que puede utilizar para validar dispositivos IoT para lograr una conectividad fiable y segura con AWS IoT Core, antes de desplegar dispositivos en producción. Las pruebas prediseñadas de Device Advisor le ayudan a validar el software del dispositivo con las prácticas recomendadas para el uso de [TLS](#), [MQTT](#), [Sombra de dispositivos](#), y [Trabajos de IoT](#). También puede descargar informes de calificación firmados para enviarlos a la AWS Partner Network para que su dispositivo califique para el [AWS Catálogo de dispositivos asociados](#) sin necesidad de enviar su dispositivo y esperar a que se pruebe.

Note

El asesor de dispositivos se admite en las regiones us-east-1, us-west-2, ap-northeast-1 y eu-west-1. Device Advisor admite MQTT con certificados de cliente X509.

El capítulo contiene las siguientes secciones:

- [Configuración \(p. 1058\)](#)
- [Introducción a Device Advisor en la consola \(p. 1062\)](#)
- [Flujo de trabajo del Advisor \(p. 1068\)](#)
- [Flujo de trabajo de consola detallado de Device \(p. 1072\)](#)
- [Casos de prueba de Device Advisor \(p. 1082\)](#)

Cualquier dispositivo que se haya creado para conectarse a AWS IoT Core puede aprovechar Device Advisor. Puede acceder a Device Advisor desde la [AWS IoT consola](#), o mediante el AWS CLI o SDK de. Cuando esté listo para probar su dispositivo, regístrelo con AWS IoT Core y configure el software del dispositivo con el endpoint de Device Advisor. A continuación, elija las pruebas precompiladas, configúrelas, ejecute las pruebas en su dispositivo y obtenga los resultados de las pruebas junto con registros detallados o un informe de calificación.

Device Advisor es un punto final de prueba en el AWSnube. Puede probar los dispositivos configurándolos para que se conecten al punto final de prueba proporcionado por el Asesor de dispositivos. Una vez configurado un dispositivo para conectarse al punto final de prueba, puede visitar la consola del Device Advisor o utilizar el AWS SDK para elegir las pruebas que desea ejecutar en sus dispositivos. A continuación, Device Advisor administra todo el ciclo de vida de una prueba, incluido el aprovisionamiento de recursos, la programación del proceso de prueba, la administración de la máquina de estado, el registro del comportamiento del dispositivo, el registro de los resultados y la entrega de los resultados finales en forma de informe de prueba.

Configuración

Antes de usar Device Advisor por primera vez, realice las siguientes tareas.

Requisitos previos

- [Cree una objeto de IoT \(p. 1059\)](#)
- [Crear un rol de IAM para utilizarlo como rol de dispositivo \(p. 1059\)](#)
- [Crear una política administrada personalizada para que un usuario de IAM utilice Device Advisor \(p. 1060\)](#)

- [Crear un usuario de IAM para usar Device Advisor \(p. 1061\)](#)
- [Configuración del dispositivo \(p. 1061\)](#)

Cree una objeto de IoT

Primero tendrá que crear una cosa y asociar un certificado al objeto. Utilice el siguiente tutorial para crear una cosa: [Creación de un objeto objeto](#).

Crear un rol de IAM para utilizarlo como rol de dispositivo

Note

Puede crear rápidamente el rol de dispositivo mediante la consola de Device Advisor. Consulte [Introducción al asesor de dispositivos en la consola](#) para conocer los pasos para configurar el rol de dispositivo mediante la consola de Device Advisor.

1. Vaya a [AWSConsola de IAM](#) e inicie sesión en la cuenta que utiliza para las pruebas de Device Advisor.
2. En el panel de navegación izquierdo, elija **Políticas**.
3. Elija **Create Policy** (Crear política).
4. UNDERCrear política, realice una de las siguientes opciones:
 1. Para **Service** (Servicio), eligeloT.
 2. UNDERActions, seleccione acciones basadas en la política adjunta a la cosa de IoT o el certificado creado en la sección anterior (recomendado) o busque las siguientes acciones en el Filtrocuadro de acción y selecciónalos.
 - Conectar
 - Publicación
 - Suscribirse
 - Recibir
 3. UNDERRecursos, o prácticas recomendadas de seguridad, le recomendamos que restrinja el cliente, Tema de, y filtro de tema recursos mediante los siguientes pasos:
 - a. ElegirEspecificar ARN de recursos cliente para la acción Connect.
 - i. Elija Add ARN (Agregar ARN).
 - ii. Especifique la `region`, `accountId`, y `clientId` en el editor ARN visual o especifique manualmente los nombres de recursos de Amazon (ARN) de los temas de IoT que desea utilizar para ejecutar casos de prueba. La `clientId` del dispositivo MQTTclientID del dispositivo utiliza para interactuar con Device Advisor.
 - iii. Elija Add.
 - b. ElegirEspecificar ARN de recursos de tema para la acción Recibir y 1 más.
 - i. Elija Add ARN (Agregar ARN).
 - ii. Especifique la `region`, `accountId`, y `Nombre del tema` en el editor de ARN visual o especifique manualmente los ARN de los temas de IoT que desea utilizar para ejecutar casos de prueba. La `Nombre del tema` del dispositivo MQTTTopic del dispositivo utiliza para publicar mensajes en.
 - iii. Elija Add.
 - c. ElegirEspecificar ARN de recursos de filtro de temas para la acción Suscribirse.
 - i. Elija Add ARN (Agregar ARN).
 - ii. Especifique la `region`, `accountId`, y `Nombre del tema` en el editor de ARN visual o especifique manualmente los ARN de los temas de IoT que desea utilizar para ejecutar casos de prueba. La `Nombre del tema` del dispositivo MQTTTopic del dispositivo utiliza para suscribirse.

- iii. Elija Add (Aregar).
5. Elija Review policy (Revisar política).
 6. UNDERPolítica de revisión, introduzca unNombre.
 7. Elija Create Policy (Crear política).
 8. En el panel de navegación izquierdo, elijaRoles de.
 9. Elija Create Role (Crear rol).
 10. UNDERO seleccione un servicio para ver sus casos de uso, eligeloT.
 11. En Select your use case (Seleccione su caso de uso), elija IoT.
 12. Seleccione Next (Siguiente): Permisos.
 13. (Opcional) EnDefinir límite de permisos, ElijaUtilice un límite de permisos para controlar los permisos que puede tener el rol como máximoy, a continuación, elija la política que acaba de crear.
 14. Seleccione Next (Siguiente): Tags (Etiquetas).
 15. Seleccione Next (Siguiente): Consulte.
 16. Escriba unNombre del roly unDescripción del rol.
 17. Elija Create role (Crear rol).
 18. Desplácese hasta el rol que ha creado.
 19. En el navegadorPermisospestaña, elijaAsociar políticasy, a continuación, elija la política que ha creado en el paso 4
 20. Elija Asociar política.
 21. ElegirRelaciones de confianzay seleccioneModificar relación de confianza.
 22. Introduzca esta política:

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

23. Elija Update Trust Policy (Actualizar política de confianza).

Crear una política administrada personalizada para que un usuario de IAM utilice Device Advisor

1. Desplácese hasta la consola de IAM en<https://console.aws.amazon.com/iam/>. Si se le solicita, escriba suAWS Credenciales de para iniciar sesión.
2. En el panel de navegación izquierdo, elija Policies (Políticas).
3. Elegir Crear política y luego seleccione JSON pestaña.
4. Añada los permisos necesarios para utilizar Device Advisor. El documento de política se encuentra en el temaPrácticas recomendadas de seguridad.
5. Elija Review Policy (Revisar la política).
6. Introduzca un Name (Nombre) y una Description (Descripción).
7. Elija Create Policy (Crear política).

Crear un usuario de IAM para usar Device Advisor

Note

Le recomendamos que cree un usuario de IAM para utilizarlo cuando ejecute pruebas de Device Advisor. No se recomienda utilizar un usuario administrador de IAM para ejecutar pruebas de Device Advisor, si está permitido.

1. Desplácese hasta la consola de IAM en <https://console.aws.amazon.com/iam/>. Si se le solicita, escriba suAWS Credenciales de para iniciar sesión.
2. En el panel de navegación izquierdo, Elija Usuarios de.
3. Elija Add user (Agregar usuario).
4. Escriba unUser name (Nombre de usuario):.
5. Select Acceso programático.
6. Seleccione Next (Siguiente): Permisos.
7. Elija Attach existing policies directly.
8. Introduzca el nombre de la política administrada de forma personalizada que creó en el cuadro de búsqueda y, a continuación, seleccione la casilla de verificación para Nombre de la política.
9. Seleccione Next (Siguiente): Tags (Etiquetas).
10. Seleccione Next (Siguiente): Consulte.
11. Seleccione la opción Create user (Crear usuario).
12. Elija Close (Cerrar).

Device Advisor requiere acceso a suAWS Recursos (cosas, certificados y puntos finales) en su nombre. El usuario de IAM debe tener los permisos necesarios. Device Advisor también publicará registros en Amazon CloudWatch si adjunta la política de permisos necesaria al usuario de IAM.

Configuración del dispositivo

Device Advisor utiliza la extensión TLS de indicación de nombre de servidor (SNI) para aplicar configuraciones TLS. Los dispositivos deben usar esta extensión al conectarse y pasar un nombre de servidor idéntico al punto final de prueba de Device Advisor.

Device Advisor permite la conexión TLS cuando se realiza una prueba en elRunningy deniega la conexión TLS antes y después de cada prueba. Por este motivo, también recomendamos utilizar el mecanismo de reinicio de conexión de dispositivos para tener una experiencia de prueba totalmente automatizada con Device Advisor. Si ejecuta un conjunto de pruebas con más de un caso de prueba (por ejemplo, TLS connect, MQTT connect y MQTT publish), le recomendamos que tenga un mecanismo creado para que su dispositivo intente conectarse a nuestro punto final de prueba cada cinco segundos. A continuación, puede ejecutar varios casos de prueba, en secuencia y de forma automatizada.

Note

Para que el software del dispositivo esté listo para su prueba, le recomendamos que tenga un SDK que pueda conectarse a AWS IoT Core y actualice el SDK con el extremo de prueba de Device Advisor proporcionado para su cuenta.

Device Advisor admite dos tipos de puntos de enlace: Endpoints a nivel de cuenta y endpoints a nivel de dispositivo. Elija el punto de enlace que mejor se adapte a su caso de uso. Para ejecutar simultáneamente varios conjuntos de pruebas utilizando distintos dispositivos, utilice un extremo a nivel de dispositivo. Ejecute el siguiente comando para obtener el punto de enlace de nivel de dispositivo:

```
aws iotdeviceadvisor get-endpoint --thing-arn your-thing-arn
```

o bien

```
aws iotdeviceadvisor get-endpoint --certificate-arn your-certificate-arn
```

Para ejecutar un conjunto de pruebas a la vez, elija un punto final a nivel de cuenta. Ejecute el siguiente comando para obtener el punto de enlace de nivel de cuenta:

```
aws iotdeviceadvisor get-endpoint
```

Introducción a Device Advisor en la consola

Este tutorial le ayuda a comenzar rápidamente a utilizar Device Advisor en la consola. Device Advisor ofrece características tales como pruebas obligatorias e informes de calificación firmados para calificar y enumerar dispositivos en el [AWS Catálogo de dispositivos asociados](#) tal como se detalla en el [AWS IoT Core Programa de cualificación](#).

Para obtener más información acerca del uso de Device Advisor, consulte [Flujo de trabajo del Advisor \(p. 1068\)](#) y [Flujo de trabajo de consola detallado de Device \(p. 1072\)](#).

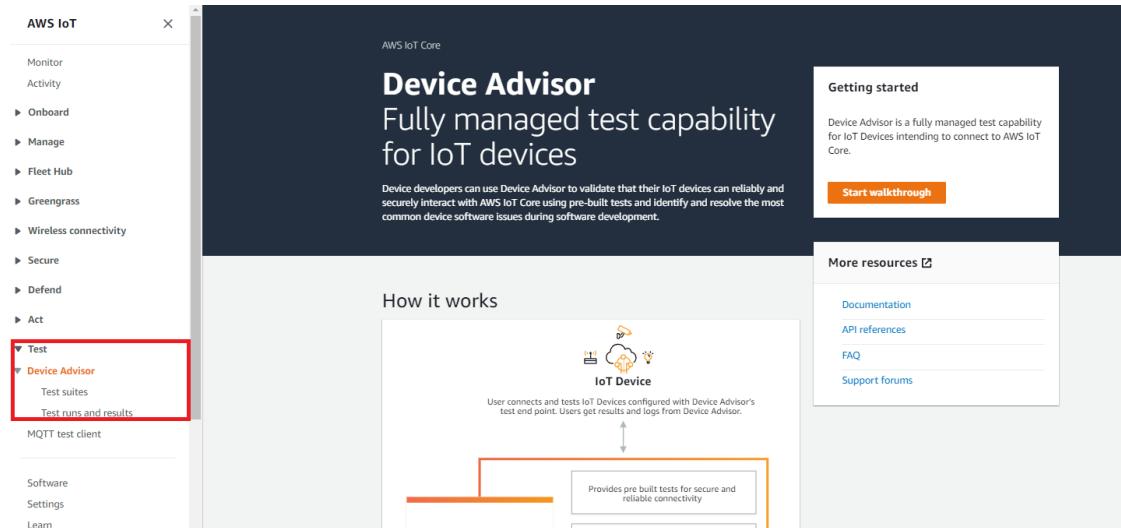
Para completar este tutorial, siga los pasos que se describen en [Configuración \(p. 1058\)](#).

Note

El asesor de dispositivos se admite en las regiones us-east-1, us-west-2, ap-northeast-1 y eu-west-1.

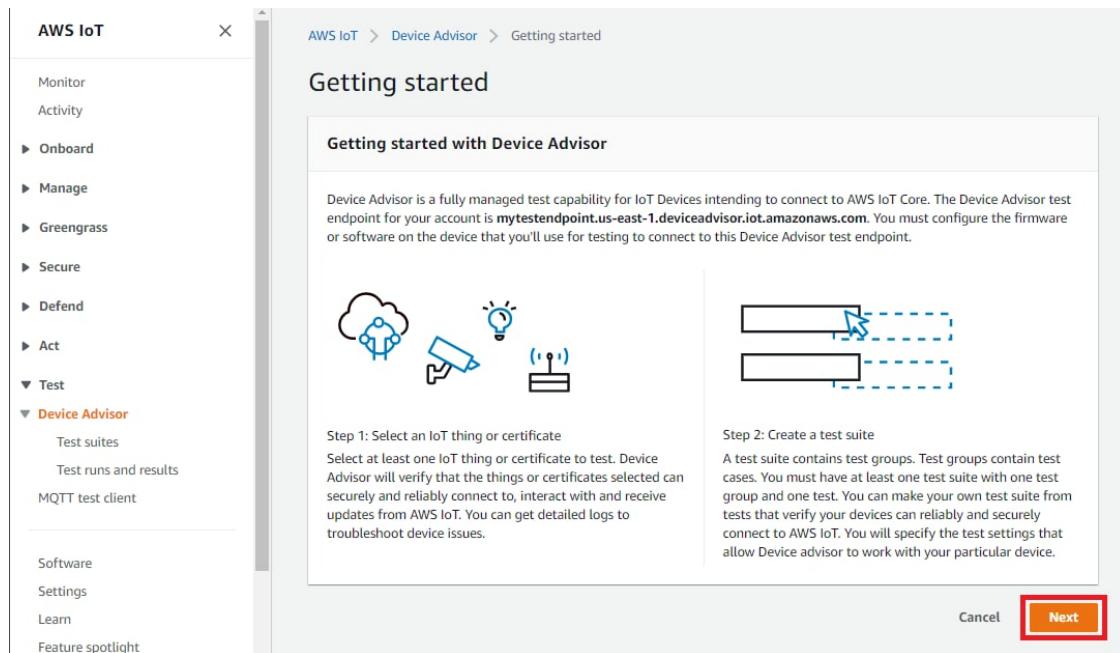
Introducción

1. En el navegador [AWS IoT consola](#), en el panel de navegación, expanda Pruebas, y luego Device Advisor y luego iniciar tutorial.



2. La [Introducción a Device Advisor](#) ofrece una descripción general de los pasos necesarios para crear un conjunto de pruebas y ejecutar pruebas en su dispositivo. También puede encontrar el punto de enlace de prueba de Device Advisor de su cuenta. Debe configurar el firmware o el software en el dispositivo que utilizará para realizar pruebas para conectarse a este extremo de prueba.

Necesitará que para completar este tutorial [Crear un objeto y certificado](#).



Una vez que hayas revisado la información, eligePróximo.

3. EnPaso 1, seleccione unAWS IoTcosa o certificado para probar con Device Advisor. Si no tiene ningún elemento o certificado existente, consulte[Configuración de](#).

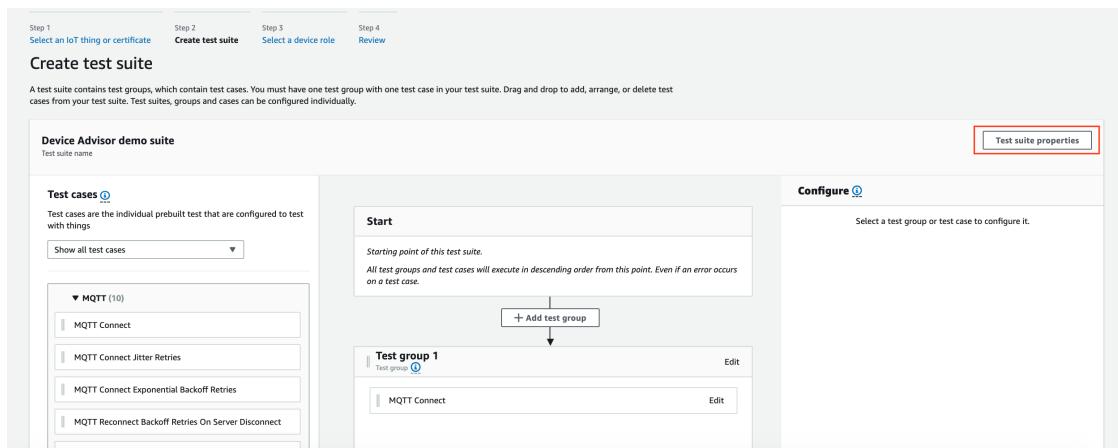
En el navegadorPunto de enlace de prueba, seleccione el punto final que mejor se adapte a nuestro caso de uso. Si planea ejecutar varios conjuntos de pruebas simultáneamente utilizando el mismoAWS cuenta, seleccionePunto final a nivel de dispositivo. De lo contrario, para ejecutar un conjunto de pruebas a la vez, seleccioneEndpoint de nivel de cuenta.

Configure el dispositivo de prueba con el punto final de prueba del Asesor de dispositivos seleccionado.

Elija Next (Siguiente).

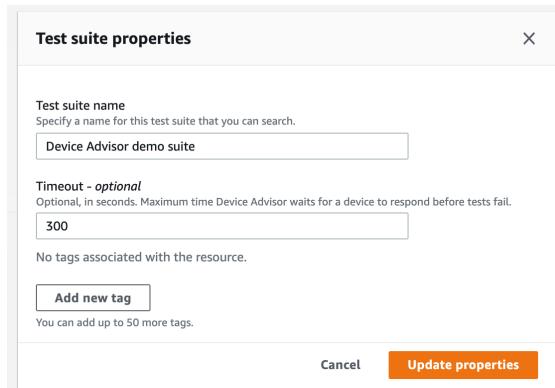
4. EnPaso 2, crea y configura un conjunto de pruebas personalizado. Un conjunto de pruebas personalizado debe tener al menos un grupo de prueba y cada grupo de prueba debe tener al menos un caso de prueba. Hemos añadido elConexión MQTTcaso de prueba para que empieces.

ElegirPropiedades del conjunto de pruebas. Debe proporcionar las propiedades del conjunto de pruebas al crear el conjunto de pruebas.



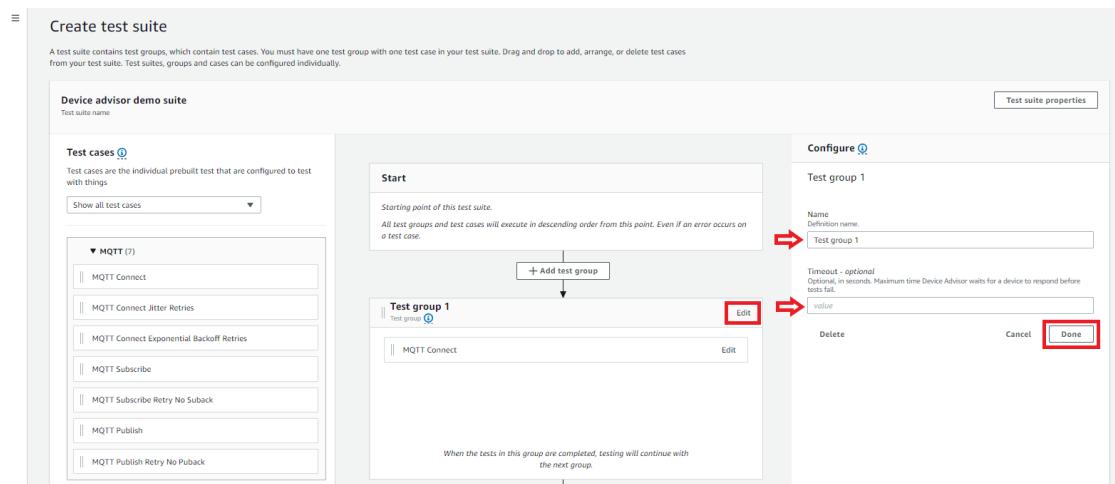
Puede configurar las siguientes propiedades de nivel de suite:

- Nombre del conjunto de pruebas: Introduzca un nombre para el grupo de prueba.
- Timeout (Tiempo de espera):(opcional): Tiempo de espera en segundos para cada caso de prueba del conjunto de pruebas actual. Si no especifica un valor de tiempo de espera, se utiliza el valor predeterminado.
- Etiquetas(opcional): Añade etiquetas al conjunto de pruebas que vas a crear.



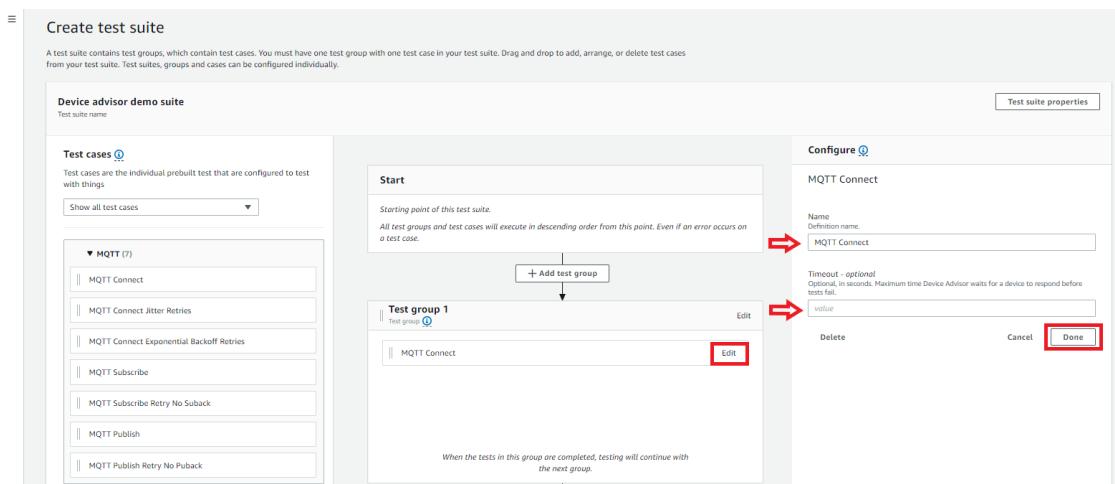
Cuando hayas terminado, eligeActualizar propiedades.

5. (Opcional) Puede actualizar la configuración del grupo de conjuntos de pruebas seleccionandoEditarjunto al nombre del grupo de prueba.
 - Nombre: Introduzca un nombre personalizado para el grupo de pruebas.
 - Timeout (Tiempo de espera):(opcional): Tiempo de espera en segundos para cada caso de prueba del conjunto de pruebas actual. Si no especifica un valor de tiempo de espera, se utiliza el valor predeterminado.



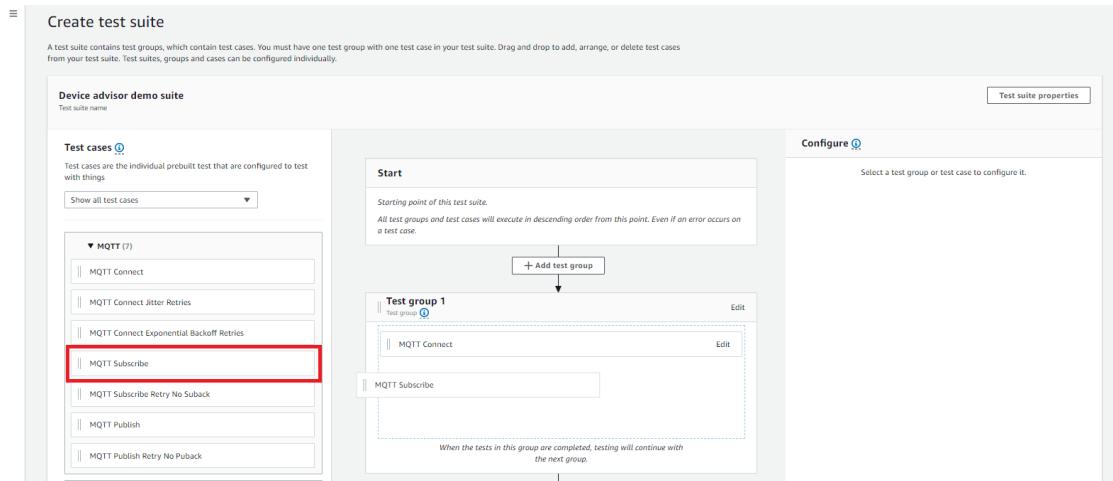
Seleccione Done (Listo).

6. (Opcional) Puede actualizar la configuración del caso de prueba seleccionando Editar junto al nombre del caso de prueba.
 - Nombre: Introduzca un nombre personalizado para el grupo de pruebas.
 - Timeout (Tiempo de espera):(opcional): Tiempo de espera en segundos para el caso de prueba seleccionado. Si no especifica un valor de tiempo de espera, se utiliza el valor predeterminado.



Seleccione Done (Listo).

7. (Opcional) Para añadir más grupos de pruebas al conjunto de pruebas, elija Añadir grupo de pruebas y siga las instrucciones del paso 5.
8. (Opcional) Para añadir más casos de prueba, arrastre los casos de prueba que se muestran en Casos de prueba en cualquier grupo de prueba.



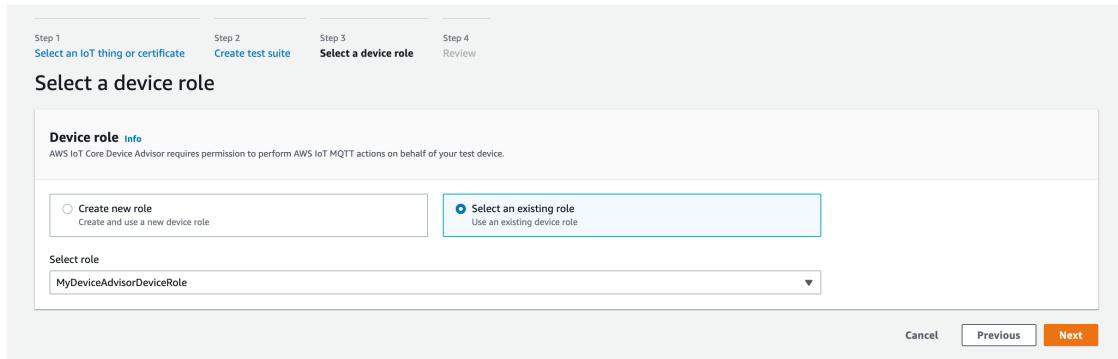
- Los grupos de prueba y los casos de prueba se pueden reordenar seleccionando y arrastrando los casos de prueba enumerados. Device Advisor ejecuta las pruebas en el orden en que se enumeran los casos de prueba.

Después de configurar el conjunto de pruebas, elija Próximo.

- En Paso 3 puede configurar un rol de dispositivo que Device Advisor utilizará para realizar AWS IoT Acciones de MQTT en nombre de su dispositivo de prueba. Si seleccionó el Conexión MQTT caso de prueba únicamente, el Conectarse seleccionará automáticamente, ya que se requiere ese permiso en el rol del dispositivo para ejecutar este conjunto de pruebas. Si ha seleccionado otros casos de prueba, se seleccionarán las acciones correspondientes.

Proporcione los valores de recurso para cada una de las acciones seleccionadas. Por ejemplo, para el Conectar, proporcione el identificador de cliente con el que se conectará el dispositivo al endpoint de Device Advisor. Puede proporcionar varios valores mediante comas para separar los valores y puede proporcionar valores de prefijo mediante un carácter comodín (*). Por ejemplo, para proporcionar permiso para publicar sobre cualquier tema que empiece por MyTopic, puede proporcionar »MyTopic*» como valor de recurso.

Si ya ha creado un rol de dispositivo desde [Configuración de](#), y desea usar ese rol, elija [Selección de una función existente](#) y elige la función de tu dispositivo en [Selección de una función](#).



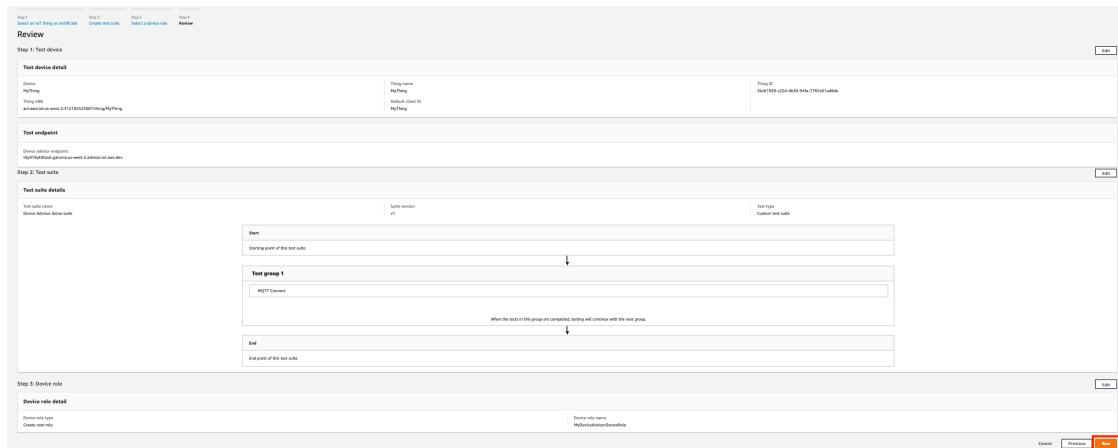
Configure el rol de su dispositivo utilizando una de las dos opciones proporcionadas y elija Próximo.

11. Paso 4 muestra una descripción general del dispositivo de prueba seleccionado, el punto final de prueba, el conjunto de pruebas y la función de dispositivo de prueba que ha configurado. Si desea realizar cambios en una sección, elija la Edición de la sección que quieras editar.

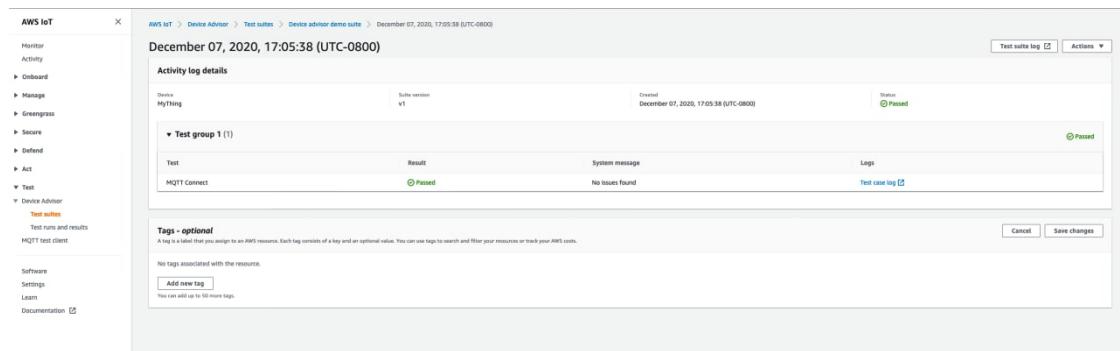
Note

Para obtener los mejores resultados, puede conectar el dispositivo de prueba seleccionado al extremo de prueba de Device Advisor antes de iniciar la ejecución del conjunto de pruebas. Le recomendamos que tenga un mecanismo creado para que su dispositivo intente conectarse a nuestro punto final de prueba cada cinco segundos durante uno o dos minutos.

Para crear el conjunto de pruebas y ejecutar las pruebas seleccionadas en su dispositivo, elija Ejecución de.



12. En el panel de navegación expanda Pruebas, Device Advisory luego Ejecuciones de pruebas y resultados para ver los detalles de la ejecución y los registros. Seleccione la ejecución del conjunto de pruebas que comenzó para ver los detalles de la ejecución y los registros.



13. Para obtener acceso a CloudWatch registros para la ejecución de la suite:
 - Elegir Registro del conjunto de pruebas para ver el CloudWatch registros de ejecución del grupo de prueba.
 - Elegir Registro de casos de pruebas para cualquier caso de prueba para ver un caso específico CloudWatch registros.
14. En función de los resultados de las pruebas,[solucionar problemas](#)El dispositivo hasta que se superan todas.

Flujo de trabajo del Advisor

En este tutorial se proporcionan instrucciones sobre cómo crear un conjunto de pruebas personalizado y ejecutar pruebas en el dispositivo que desea probar en la consola. Una vez completadas las pruebas, puede ver los resultados de las pruebas y los registros detallados.

Tutoriales

- [Requisitos previos \(p. 1068\)](#)
- [Creación de una definición de conjunto de pruebas \(p. 1068\)](#)
- [Obtenga una definición de conjunto de pruebas \(p. 1070\)](#)
- [Obtener un punto final de prueba \(p. 1070\)](#)
- [Iniciar una ejecución de un conjunto de pruebas \(p. 1071\)](#)
- [Ejecute un conjunto de pruebas \(p. 1071\)](#)
- [Detener la ejecución de un conjunto de pruebas \(p. 1071\)](#)
- [Obtenga un informe de calificación para que se ejecute correctamente el conjunto de pruebas de calificación \(p. 1072\)](#)

Requisitos previos

Para completar este tutorial, complete los pasos que se describen en[Configuración \(p. 1058\)](#).

Creación de una definición de conjunto de pruebas

En primer lugar,[Instale un AWSSDK](#).

Sintaxis de rootGroup

Un grupo raíz es una cadena JSON que especifica qué casos de prueba se incluyen en el conjunto de pruebas, así como las configuraciones necesarias para esos casos de prueba. Utilice el grupo raíz para

estructurar y ordenar su conjunto de pruebas de la forma que desee. La jerarquía de un conjunto de pruebas es:

```
test suite # test group(s) # test case(s)
```

Un grupo de pruebas debe tener al menos un grupo de pruebas y cada grupo de prueba debe tener al menos un caso de prueba. Device Advisor ejecuta las pruebas en el orden en que se definen los grupos de prueba y los casos de prueba.

Cada grupo raíz sigue esta estructura 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": ""
          }
        }
      ]
    }
  ]
}
```

Un bloque que contiene un "name", "configuration", y "tests" se denomina «definición de grupo». Un bloque que contiene un "name", "configuration", y "test" se denomina «definición de caso de prueba». Cada "test" bloque que contiene un "id" y "version" se denomina «caso de prueba».

Para obtener información sobre cómo rellenar el "id" y "version" campos para cada caso de ensayo ("test" block), consulte [Caso de prueba de Device Advisor \(p. 1082\)](#). En esa sección también se incluye información sobre los "configuration" Configuración del .

El siguiente bloque es un ejemplo de configuración de grupo raíz que especifica los casos de prueba «MQTT Connect Happy Case» y «Reintentos de retroceso exponencial de MQTT Connect», junto con descripciones de los campos de configuración.

```
{
  "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
    }
}
]
```

Debe proporcionar la configuración del grupo raíz al crear la definición del conjunto de pruebas. Save the `suiteDefinitionId` que se devuelve en el objeto de respuesta. Este ID se utiliza para recuperar la información de definición del conjunto de pruebas y para ejecutar el conjunto de pruebas.

Este es un ejemplo del SDK de Java:

```
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()
)
```

Obtenga una definición de conjunto de pruebas

Después de iniciar la ejecución de un conjunto de pruebas, puede comprobar su progreso y sus resultados con el `GetSuiteRunAPI`.

Ejemplo del SDK de

```
// Using the SDK, call the GetSuiteRun API.

response = iotDeviceAdvisorClient.GetSuiteRun(
    GetSuiteRunRequest.builder()
        .suiteDefinitionId("your-suite-definition-id")
        .suiteRunId("your-suite-run-id")
    .build()
```

Obtener un punto final de prueba

Puede usar `GetTestEndpointAPI` para obtener el punto final de prueba utilizado por su dispositivo. Al elegir el punto final, seleccione el punto final que mejor se ajuste a la situación. Para ejecutar simultáneamente varios conjuntos de pruebas, utilice endpoint a nivel de dispositivo proporcionando

unthing ARN o uncertificate ARN. Para ejecutar un único conjunto de pruebas, elija el punto final de nivel de cuenta sin proporcionar argumentos.

Ejemplo del SDK de

```
response = iotDeviceAdvisorClient.getEndpoint(GetEndpointRequest.builder()
    .certificateArn("your-test-device-certificate-arn")
    .thingArn("your-test-device-thing-arn")
    .build())
```

Iniciar una ejecución de un conjunto de pruebas

Después de crear correctamente una definición de conjunto de pruebas y configurar el dispositivo de prueba para conectarse al punto final de prueba de Device Advisor, ejecute el conjunto de pruebas con elStartSuiteRunAPI. Utilice cualquiera de lascertificateArnnothingArnpara ejecutar el conjunto de pruebas. Si ambos están configurados, el certificado se utilizará si pertenece a la cosa.

Para.parallelRun(), use true si utiliza endpoint de nivel de dispositivo para ejecutar varios conjuntos de pruebas en parallel mediante unoAWSaccount.

Ejemplo del SDK de

```
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())
```

Save the suiteRunId que se devuelve en la respuesta. Lo utilizará para recuperar los resultados de la ejecución de este conjunto de pruebas.

Ejecute un conjunto de pruebas

Después de crear la definición de su conjunto de pruebas, recibirá el suiteDefinitionId en el objeto de respuesta delCreateSuiteDefinitionAPI.

Puede que veas que hay nuevos id campos dentro de cada uno de los grupos y definiciones de casos de prueba del grupo raíz que se devuelve. Esto se espera; puede utilizar estos ID para ejecutar un subconjunto de la definición de su conjunto de pruebas.

Ejemplo del SDK de Java:

```
response = iotDeviceAdvisorClient.GetSuiteDefinition(
    GetSuiteDefinitionRequest.builder()
        .suiteDefinitionId("your-suite-definition-id")
        .build()
)
```

Detener la ejecución de un conjunto de pruebas

Para detener la ejecución de un conjunto de pruebas que aún está en curso, puede llamar alStopSuiteRunAPI. Después de llamar alStopSuiteRunAPI, el servicio iniciará el proceso de limpieza.

Mientras el servicio ejecuta el proceso de limpieza, el estado de ejecución del conjunto de pruebas se actualiza a `astopping`. El proceso de limpieza durará varios minutos y, una vez finalizado el proceso, el estado de ejecución del conjunto de pruebas se actualiza a `astopped`. Una vez que una ejecución de prueba se haya detenido por completo, podrá iniciar otro conjunto de pruebas. Puede comprobar periódicamente el estado de ejecución de la suite mediante la `GetSuiteRunAPI` como se muestra en la sección anterior.

Ejemplo del SDK de

```
// Using the SDK, call the StopSuiteRun API.  
  
response = iotDeviceAdvisorClient.StopSuiteRun(  
    StopSuiteRun.builder()  
        .suiteDefinitionId("your-suite-definition-id")  
        .suiteRunId("your-suite-run-id")  
    .build())
```

Obtenga un informe de calificación para que se ejecute correctamente el conjunto de pruebas de calificación

Si ejecuta un conjunto de pruebas de calificación que se completa, puede recuperar un informe de calificación mediante el `GetSuiteRunReportAPI`. Puede utilizar este informe de calificación para calificar su dispositivo con el AWS IoT Core Programa de calificación. Para determinar si el grupo de pruebas es un conjunto de pruebas de calificación, compruebe si el `intendedForQualification` establece `true`. Después de llamar al `GetSuiteRunReportAPI`, la URL de descarga devuelta está disponible para que pueda descargarla durante 90 segundos. Si transcurren más de 90 segundos de la vez anterior que llamó al `GetSuiteRunReportAPI`, vuelve a llamar a la API para recuperar una URL válida.

Ejemplo del SDK de

```
// Using the SDK, call the getSuiteRunReport API.  
  
response = iotDeviceAdvisorClient.getSuiteRunReport(  
    GetSuiteRunReportRequest.builder()  
        .suiteDefinitionId("your-suite-definition-id")  
        .suiteRunId("your-suite-run-id")  
    .build())
```

Flujo de trabajo de consola detallado de Device

En este tutorial, creará un conjunto de pruebas personalizado y ejecutará pruebas con el dispositivo que desea probar en la consola. Una vez completadas las pruebas, puede ver los resultados de las pruebas y los registros detallados.

Tutoriales

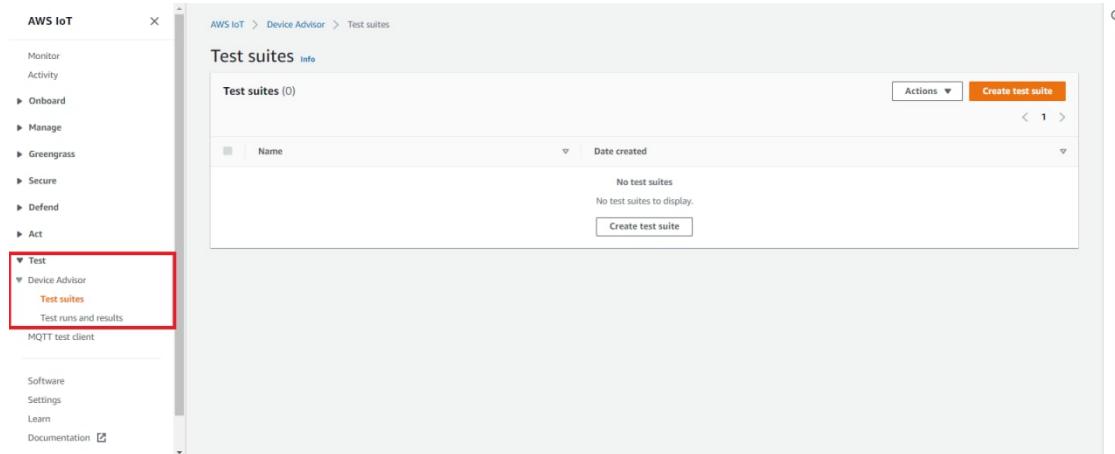
- [Requisitos previos \(p. 1073\)](#)
- [Creación de una definición de conjunto de pruebas \(p. 1073\)](#)
- [Iniciar una ejecución de un conjunto de pruebas \(p. 1077\)](#)
- [Detener la ejecución de un conjunto de pruebas \(opcional\) \(p. 1079\)](#)
- [Ver detalles y registros de ejecución del conjunto de pruebas \(p. 1081\)](#)
- [Descarga de AWS IoT Informe de calificación \(p. 1082\)](#)

Requisitos previos

Necesitará que para completar este tutorial [Crear un objeto y certificado](#).

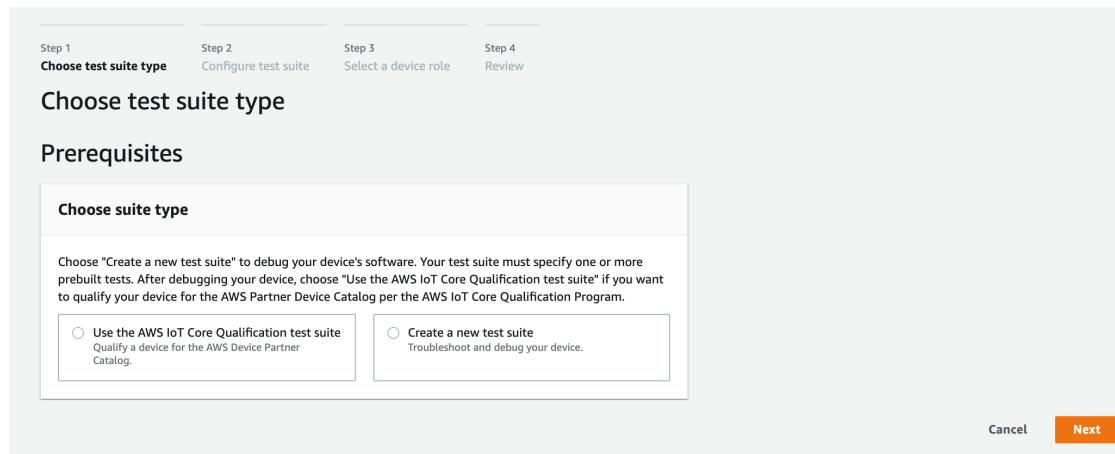
Creación de una definición de conjunto de pruebas

1. En el navegador [AWS IoT consola](#), en el panel de navegación, expanda Pruebas, Device Advisory luego Conjuntos de pruebas.



Elegir [Crear conjunto de pruebas](#).

2. Seleccione cualquiera [Use the AWS Qualification test suite](#) o [Create a new test suite](#).



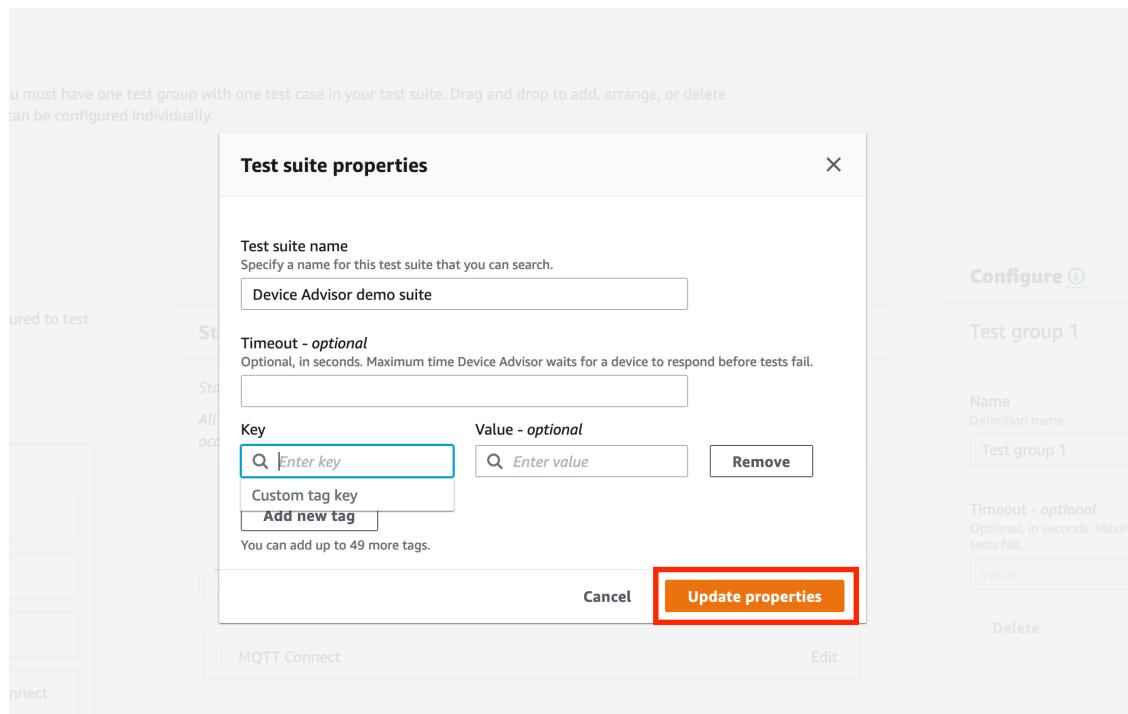
Select [Use the AWS Qualification test suite](#) para calificar y publicar tu dispositivo en el AWS Catálogo de dispositivos de socios. Al elegir esta opción, se requieren casos de prueba para la calificación de su dispositivo en el AWS IoT Core el programa de calificación está preseleccionado. Los grupos de prueba y los casos de prueba no se pueden agregar ni eliminar. Necesitará configurar las propiedades del conjunto de pruebas.

Select [Create a new test suite](#) para crear y configurar un conjunto de pruebas personalizado. Recomendamos comenzar con esta opción para las pruebas iniciales y la solución de problemas. Un conjunto de pruebas personalizado debe tener al menos un grupo de prueba y cada grupo de prueba debe tener al menos un caso de prueba. A los efectos de este tutorial, seleccionaremos esta opción y seleccionaremos [Próximo](#).

- Elegir Propiedades del conjunto de pruebas. Debe crear las propiedades del conjunto de pruebas al crear el conjunto de pruebas.

UNDERPropiedades del conjunto de pruebas, rellene lo siguiente:

- Nombre del conjunto de pruebas: Puede crear la suite con un nombre personalizado.
- Timeout (Tiempo de espera):(opcional): Tiempo de espera en segundos para cada caso de prueba del conjunto de pruebas actual. Si no especifica un valor de tiempo de espera, se utiliza el valor predeterminado.
- Etiquetas(opcional): Agregar etiquetas al grupo de prueba.



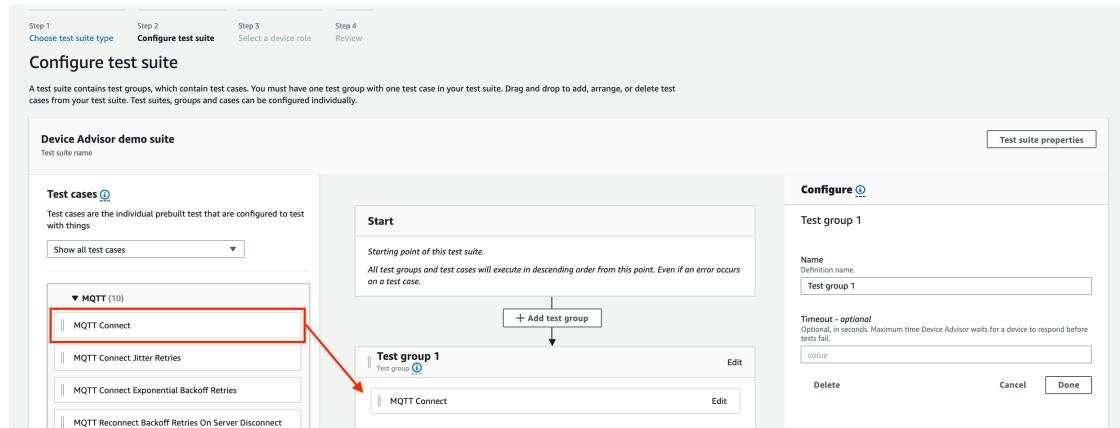
Cuando haya terminado, elija Actualizar propiedades.

- Para modificar la configuración de nivel de grupo, en **Test group 1**, elige **Editar**. A continuación, introduzca un nombre para asignar al grupo un nombre personalizado.

Si lo desea, también puede escribir un **Timeout** (Tiempo de espera):valor en segundos en el grupo de prueba seleccionado. Si no especifica un valor de tiempo de espera, se utiliza el valor predeterminado.

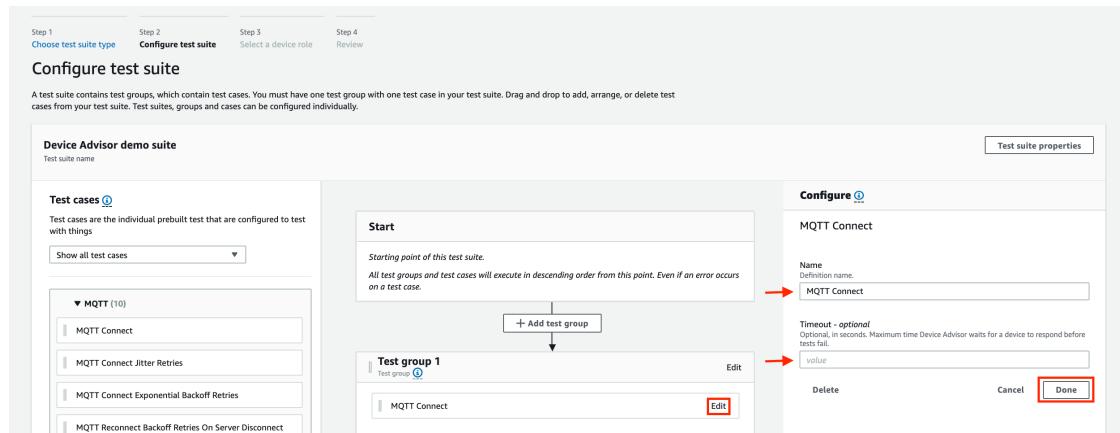
Seleccione Done (Listo).

- Arrastre uno de los casos de prueba disponibles desde Casos de prueba en el grupo de pruebas.



- Para modificar la configuración de nivel de caso de prueba para el caso de prueba que ha agregado al grupo de prueba, elija **Editar**. A continuación, introduzca unNombre para asignar al grupo un nombre personalizado.

Si lo desea, también puede escribir un **Timeout (Tiempo de espera):valor** en segundos en el grupo de prueba seleccionado. Si no especifica un valor de tiempo de espera, se utiliza el valor predeterminado.



Seleccione **Done (Listo)**.

Note

Para añadir más grupos de pruebas al conjunto de pruebas, elija **Añadir grupo de pruebas**. Siga los pasos anteriores para crear y configurar más grupos de pruebas o agregar más casos de prueba a uno o más grupos de pruebas. Los grupos de prueba y los casos de prueba se pueden reordenar eligiendo y arrastrando un caso de prueba a la posición deseada. Device Advisor ejecuta las pruebas en el orden en que se definen los grupos de prueba y los casos de prueba.

- Elija **Next (Siguiente)**.
- En **Paso 3**, configure un rol de dispositivo que Device Advisor utilizará para ejecutar **AWS IoT Acciones de MQTT** en nombre de su dispositivo de prueba.

Si seleccionó **Conección MQTT** como caso de prueba solo en **Paso 2**, el **Conectarse** comprobará automáticamente, ya que ese permiso es necesario en el rol del dispositivo para ejecutar este conjunto de pruebas. Si ha seleccionado otros casos de prueba, se comprobarán las acciones necesarias correspondientes. Asegúrese de que se proporcionan los valores de los recursos para cada una de las acciones. Por ejemplo, para el **Conectarse**, proporcione el identificador de cliente con el que el dispositivo se conectará al endpoint de Device Advisor. Puede proporcionar varios valores mediante comas para

separar los valores y también puede proporcionar valores de prefijo utilizando un carácter comodín (*). Por ejemplo, para proporcionar permiso para publicar sobre cualquier tema que empiece por MyTopic, puede proporcionar »MyTopic*» como valor de recurso.

Action	Resource type	Resource
<input checked="" type="checkbox"/> Connect	ClientID	MyClient1
<input type="checkbox"/> Publish	Topic	Specify topics to publish to, e.g. MyTopic, MyTopic*
<input type="checkbox"/> Subscribe	TopicFilter	Specify topic filters to subscribe to, e.g. MyTopic, MyTopic*
<input type="checkbox"/> Receive	Topic	Specify topics to receive from e.g. MyTopic, MyTopic*

Si ya ha creado un rol de dispositivo anteriormente y desea utilizarlo, seleccione Seleccione una función existente y elige la función de tu dispositivo en Seleccione una función.

Select role
MyDeviceAdvisorDeviceRole

Configure el rol de su dispositivo utilizando una de las dos opciones proporcionadas y elija Próximo.

- En Paso 4, asegúrese de que la configuración proporcionada en cada uno de los pasos sea precisa. Para editar la configuración proporcionada para un paso concreto, elija Editar para el paso correspondiente.

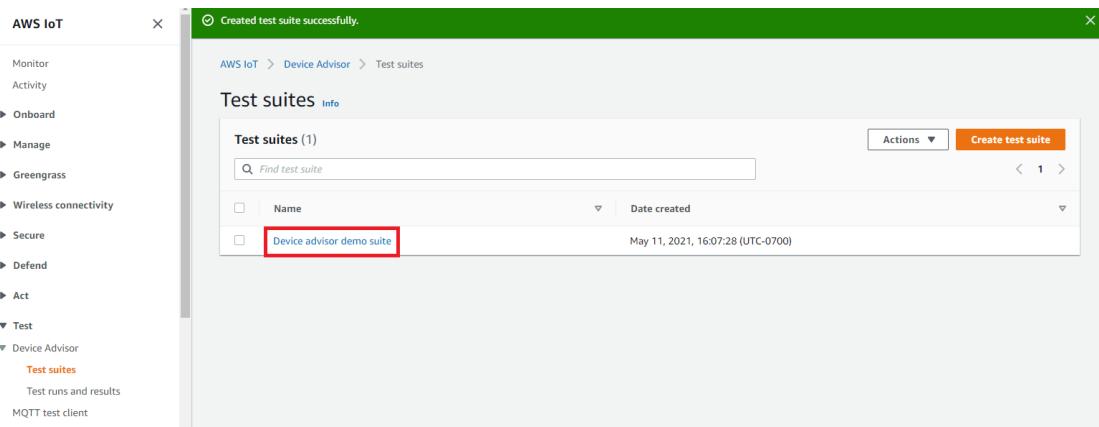
Después de verificar la configuración, elija Crear conjunto de pruebas.

El conjunto de pruebas debe crearse correctamente y se te redirigirá al Conjuntos de pruebas en la que puedes ver todo el conjunto de pruebas que se ha creado.

Si se ha producido un error en la creación del conjunto de pruebas, asegúrese de que el conjunto de pruebas, los grupos de pruebas, los casos de prueba y la función de dispositivo se hayan configurado de acuerdo con las instrucciones anteriores.

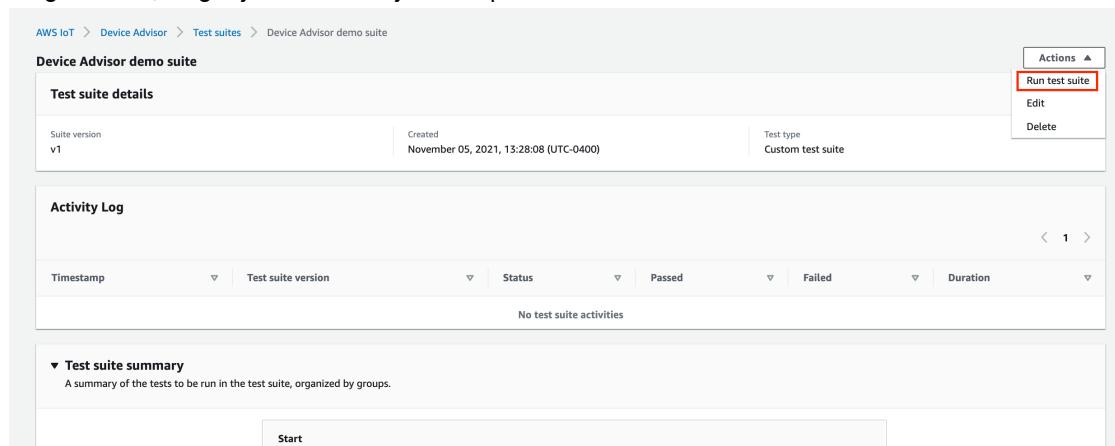
Iniciar una ejecución de un conjunto de pruebas

- En el navegador [AWS IoT consola](#), en el panel de navegación, expanda Pruebas, Device Advisory luego Conjuntos de pruebas.
- Elige el conjunto de pruebas para el que te gustaría ver los detalles del grupo de pruebas.



La página de detalles del grupo de pruebas muestra toda la información relacionada con el conjunto de pruebas.

3. ElegirActions, luegoEjecutar un conjunto de pruebas.



4. UNDERConfiguración de Ejecutar, tendrá que seleccionar unAWS IoTcosa o certificado para probar con Device Advisor. Si no tiene ningún objeto o certificado existente, primerocrearAWS IoT Corerecursos (p. 1058).

EnPunto de enlace de prueba, seleccione el punto final que mejor se ajuste a su funda. Si planea ejecutar varios conjuntos de pruebas simultáneamente utilizando el mismoAWS cuenta en el futuro, seleccioneEndpoint a nivel de dispositivo. De lo contrario, si planea ejecutar solo un conjunto de pruebas a la vez, seleccioneEndpoint de nivel de cuenta.

Configure el dispositivo de prueba con el punto final de prueba del Asesor de dispositivos seleccionado.

Después de seleccionar una cosa o certificado y elegir un punto final de Device Advisor, elijaEjecutar una prueba.

Run configuration

Select test devices

Choose the IoT thing/certificate to test using the test suite. If not listed below, you must first create a thing/certificate registered with IoT Core before you can run the test suite.

Things Choose a thing for this test suite. To create a new thing, go to IoT Things [\[?\]](#)

Certificates Choose a certificate for this test suite. To create a new certificate, go to IoT Certificates [\[?\]](#)

Things (1)

Name Type

MyThing

Test endpoint

Choose the endpoint that best fits your situation. If you want to simultaneously run multiple test suites then use 'Device-level endpoint'; if you want to run only one test suite at a time then choose the 'Account-level endpoint'.

Account-level endpoint Using this endpoint, you can only run one test suite at a time.
986dd41394a195492e5gamma.us-west-2.advisor.iot.awss.dev [\[?\]](#)

Device-level endpoint Using this endpoint you can run multiple test suites simultaneously.

Copy and paste this endpoint to your test device.

Tags - optional

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

No tags associated with the resource.

Add new tag You can add up to 50 more tags.

Cancel Run test

5. ElegirVaya a resultadosen el banner superior para ver los detalles de la ejecución de pruebas.

Device Advisor demo suite

Test suite details

Suite version v1 Created November 05, 2021, 13:40:33 (UTC-0400) Test type Custom test suite

Activity Log

Timestamp	Test suite version	Status	Passed	Failed	Duration
November 05, 2021, 13:53:23 (UTC-0400)	v1	Pending	-	-	-

Actions v1

Go to results

Detener la ejecución de un conjunto de pruebas (opcional)

1. En el navegador [AWS IoT consola](#), en el panel de navegación, expanda Pruebas, Device Advisory luego Ejecuciones de pruebas y resultados.
2. Elija el conjunto de pruebas en curso que desea detener.

AWS IoT Core Guía para desarrolladores

Detener la ejecución de un conjunto de pruebas (opcional)

Name	Timestamp	Test suite version	Status	Passed	Failed	Duration
Device Advisor demo suite	December 07, 2020, 11:16:46 (UTC-0800)	v1	In Progress	-	-	-

3. Elegir Actions, luego Conjunto de pruebas de.

4. El proceso de limpieza tardará varios minutos en completarse. Mientras se ejecuta el proceso de limpieza, el estado de ejecución de prueba será STOPPING. Espere a que finalice el proceso de limpieza y a que el estado del conjunto de pruebas cambie a STOPPED antes de iniciar una nueva ejecución de suite.

Ver detalles y registros de ejecución del conjunto de pruebas

- En el navegador [AWS IoT consola](#), en el panel de navegación, expanda Pruebas, Device Advisory luego Ejecuciones de pruebas y resultados.

En esta página se muestra:

- Número de cosas de IoT
- Número de certificados de IoT
- Número de conjuntos de pruebas en ejecución actualmente
- Todas las ejecuciones del conjunto de pruebas que se han creado

- Elija el conjunto de pruebas para el que desea ver los detalles de la ejecución y los registros.

Name	Timestamp	Test suite version	Status	Passed	Failed	Duration
Device Advisor demo suite	December 07, 2020, 11:16:46 (UTC-0800)	v1	In Progress	-	-	-

La página de resumen de la ejecución muestra el estado de la ejecución del conjunto de pruebas actual. Esta página se actualiza automáticamente cada 10 segundos. Le recomendamos que tenga un mecanismo creado para que su dispositivo intente conectarse a nuestro punto final de prueba cada cinco segundos durante uno o dos minutos. A continuación, puede ejecutar varios casos de prueba en secuencia de forma automatizada.

- Para obtener acceso a CloudWatch registros para la ejecución del conjunto de pruebas, elija Registro del conjunto de pruebas.

Para obtener acceso a CloudWatch registros para cualquier caso de prueba, elija Registro de casos de pruebas.

- En función de los resultados de las pruebas, [solucionar problemas](#) el dispositivo hasta que se superan todas.

Descarga deAWS IoTinforme de cualificación

Si eligió el UsarAWS IoTConjunto de pruebas de cualificacióndurante la creación de un grupo de pruebas y pudieron ejecutar un grupo de pruebas de calificación, puede descargar un informe de calificación eligiendo Descargar informe de cualificaciónen la página de resumen de la ejecución de pruebas.

The screenshot shows the AWS IoT Device Advisor interface. On the left, there's a navigation sidebar with options like Monitor, Activity, Onboard, Manage, greengrass, Secure, Defend, Act, Test, Device Advisor, Test suites, Software, Settings, Liam, and Documentation. The main area displays a test suite named "AWS IoT Core Qualification demo suite" from December 07, 2020, at 23:33:16 (UTC-0800). It includes sections for "Activity log details", "Device MyThing", "Suite version v1", "Created December 07, 2020, 23:33:16 (UTC-0800)", and "Status Passed". A prominent red button at the top right says "Download qualification report". Below this, there's a table titled "AWS IoT Core Qualification Program" showing test results for various categories: MQTT Connect, MQTT Subscribe, MQTT Publish, TLS Connect, TLS Insecure Server Cert, and TLS Incorrect Subject Name Server Cert. All tests show a "Passed" result with "No issues found" and a "Test case log" link. At the bottom, there's a "Tags - optional" section with a note about tags, a "Cancel" and "Save changes" button, and a "Add new tag" input field.

Casos de prueba de Device Advisor

Device Advisor proporciona pruebas predefinidas en cinco categorías.

- TLS
- Permisos y políticas
- MQTT
- Sombra
- Ejecución de Job

Note

El dispositivo necesita aprobar las siguientes pruebas para calificar según [AWSPrograma de cualificación de dispositivos](#)

- Certificado de servidor de nombres de sujeto incorrecto TLS(«Nombre común del sujeto incorrecto (CN) /Nombre alternativo del sujeto (SAN)»)
- Certificado de servidor no seguro TLS(«No firmado por CA reconocida»)
- Connect TLS(«Connect TLS»)
- Conexión MQTT(«El dispositivo envía CONNECT aAWS IoT Core(Caso feliz)»)
- Suscripción a MQTT(«Puede suscribirse (caso feliz)»)
- Publicación MQTT(«QoS0 (caso feliz)»)

TLS

Estas pruebas se utilizan para determinar si el protocolo de seguridad de la capa de transporte (TLS) entre sus dispositivos yAWS IoTes seguro.

Ruta feliz

«Connect TLS»

Valida si el dispositivo sometido a prueba puede completar el apretón de manos TLS paraAWS IoT.
Esta prueba no valida la implementación de MQTT del dispositivo cliente.

Definición de caso de prueba API:

Note

`EXECUTION_TIMEOUT`tiene un valor predeterminado de 5 minutos. Para obtener los mejores resultados, recomendamos un valor de tiempo de espera 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 Salidas del caso de prueba:

- Pasar— El dispositivo sometido a prueba completó el apretón de manos TLS conAWS IoT.
- Pase con advertencias— El dispositivo sometido a prueba completó el apretón de manos TLS conAWS IoT, pero había mensajes de advertencia TLS desde el dispositivo oAWS IoT.
- Fracasar— El dispositivo sometido a prueba no pudo completar el apretón de manos TLS conAWS IoTdebido a un error de apretón de manos.

«TLS recibe fragmentos de tamaño máximo»

Este caso de prueba le ayuda a validar si su dispositivo puede recibir y procesar fragmentos de tamaño máximo TLS. El dispositivo de prueba debe suscribirse a un tema preconfigurado con QoS 1 para recibir una gran carga útil. Puede personalizar la carga útil mediante la configuración `$ {payload}`.

Definición de caso de prueba API:

Note

`EXECUTION_TIMEOUT`tiene un valor predeterminado de 5 minutos. Para obtener los mejores resultados, recomendamos un valor de tiempo de espera 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"
    }
]
```

Conjuntos de cifrado

«Support de dispositivos TLS paraAWS IoTCipher Suites recomendadas»

Valida que los conjuntos de cifrado del mensaje de saludo del cliente TLS del dispositivo que se está probando contiene[AWS IoTConjuntos de cifrado recomendados \(p. 382\)](#). Proporciona información adicional sobre los conjuntos de cifrado compatibles con el dispositivo.

Definición de caso de prueba API:

Note

`EXECUTION_TIMEOUT`tiene un valor predeterminado de 5 minutos. Recomendamos un valor de tiempo de espera 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 Salidas del caso de prueba:

- Pasar— Los conjuntos de cifrado del dispositivo sometido a prueba contienen al menos unoAWS IoTpaquete de cifrado recomendado y no contiene ningún conjunto de cifrado no admitido.
- Pase con advertencias— Los conjuntos de cifrado de dispositivos contienen al menos unaAWS IoTconjunto de cifrado pero 1) no contiene ninguno de los conjuntos de cifrado recomendados, o 2) contienen conjuntos de cifrado no admitidos porAWS IoT. Sugerimos verificar que los conjuntos de cifrado no compatibles sean seguros.
- Fracasar— El dispositivo en los conjuntos de cifrado de prueba no contiene ninguno de losAWS IoTConjuntos de cifrado compatibles.

Certificado de servidor de mayor tamaño

«Certificado de servidor de gran tamaño TLS»

Valida si su dispositivo puede completar el apretón de manos TLS conAWS IoTrecibiendo y procesando un certificado de servidor de mayor tamaño. El tamaño del certificado de servidor (en bytes) utilizado por esta prueba es mayor que el que se utiliza actualmente en el«Connect TLS»caso de prueba y IoT Core en un 20%. Durante este caso de prueba, se probará el espacio de búfer del dispositivo para TLS. Si el espacio del búfer es lo suficientemente grande, el apretón de manos TLS se completará sin errores. Esta prueba no valida la implementación de MQTT del dispositivo. El caso de prueba finaliza una vez finalizado el proceso de apretón de manos TLS.

Definición de caso de prueba API:

Note

`EXECUTION_TIMEOUT`tiene un valor predeterminado de 5 minutos. Para obtener los mejores resultados, recomendamos un valor de tiempo de espera de 2 minutos. Si este caso de prueba falla, pero «Connect TLS»pases de casos de prueba, te recomendamos que aumentes el límite de espacio de búfer de tu dispositivo para TLS a un mayor número de bytes. Al aumentar el límite de espacio del búfer, el dispositivo puede procesar un certificado de servidor de mayor tamaño en caso de que el tamaño 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 Salidas del caso de prueba:

- Pasar— El dispositivo sometido a prueba completó el apretón de manos TLS conAWS IoT.
- Pase con advertencias— El dispositivo sometido a prueba completó el apretón de manos TLS conAWS IoT, pero hay mensajes de advertencia TLS desde el dispositivo oAWS IoT.
- Fracasar— El dispositivo sometido a prueba no pudo completar el apretón de manos TLS conAWS IoTdebido a un error durante el proceso de apretón de manos.

Certificado de servidor defectuoso

«No firmado por CA reconocida»

Valida que el dispositivo sometido a prueba cierra la conexión si se presenta con un certificado de servidor que no tiene una firma válida de la CA de ATS. Un dispositivo solo debe conectarse a un endpoint que presente un certificado válido.

Definición de caso de prueba API:

Note

`EXECUTION_TIMEOUT`tiene un valor predeterminado de 5 minutos. Recomendamos un valor de tiempo de espera 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 Salidas del caso de prueba:

- Pasar— El dispositivo sometido a prueba cerró la conexión.
- Fracasar— El dispositivo sometido a prueba completó el apretón de manos TLS conAWS IoT.

«Nombre común del sujeto (CN) /Nombre alternativo del sujeto (SAN) incorrecto»

Valida que el dispositivo sometido a prueba cierra la conexión si se presenta con un certificado de servidor para un nombre de dominio distinto del solicitado.

Definición de caso de prueba API:

Note

`EXECUTION_TIMEOUT`tiene un valor predeterminado de 5 minutos. Recomendamos un valor de tiempo de espera 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 Salidas del caso de prueba:

- Pasar— El dispositivo sometido a prueba cerró la conexión.
- Fracasar— El dispositivo sometido a prueba completó el apretón de manos TLS conAWS IoT.

Permisos y políticas

Puede utilizar las siguientes pruebas para determinar si las políticas asociadas a los certificados de sus dispositivos siguen las prácticas recomendadas estándar.

«Las directivas adjuntas de certificados de dispositivo no contienen comodines»

Valida si las políticas de permisos asociadas a un dispositivo siguen las prácticas recomendadas y no otorgan al dispositivo más permisos de los necesarios.

Definición de caso de prueba API:

Note

`EXECUTION_TIMEOUT`tiene un valor predeterminado de 1 minuto. Recomendamos establecer un tiempo de espera de al 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"
    }
]
```

MQTT

CONECTAR, DESCONECTAR y VOLVER A CONECTAR

«El dispositivo envía CONNECT aAWS IoT Core(Caso feliz)»

Valida que el dispositivo sometido a prueba envía una solicitud CONNECT.

Definición de caso de prueba API:

Note

EXECUTION_TIMEOUTtiene un valor predeterminado de 5 minutos. Recomendamos un valor de tiempo de espera de 2 minutos.

```
"tests": [
{
    "name": "my_mqtt_connect_test",
    "configuration": {
        // optional:
        "EXECUTION_TIMEOUT": "300",    // in seconds
    },
    "test": {
        "id": "MQTT_Connect",
        "version": "0.0.0"
    }
}]
```

«El dispositivo puede devolver PUBACK a un tema arbitrario para QoS1»

Este caso de prueba comprobará si el dispositivo (cliente) puede devolver un mensaje PUBACK si recibió un mensaje de publicación del agente después de suscribirse a un tema con QoS1.

Definición de caso de prueba API:

Note

EXECUTION_TIMEOUTtiene un valor predeterminado de 5 minutos. Recomendamos un valor de tiempo de espera 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"
    }
}]
```

]

«Reintentos de conexión de dispositivos con retroceso de fluctuación: sin respuesta CONNACK»

Valida que el dispositivo sometido a prueba utilice el retroceso de fluctuación adecuado al volver a conectarse con el agente durante al menos cinco veces. El agente registra la marca de hora del dispositivo bajo la solicitud CONNECT de la prueba, realiza la validación de paquetes, hace una pausa sin enviar un CONNACK al dispositivo que se está probando y espera a que el dispositivo sometido a prueba vuelva a enviar la solicitud. Se permite pasar el sexto intento de conexión y CONNACK puede regresar al dispositivo que se está probando.

El proceso anterior se vuelve a realizar. En total, este caso de prueba requiere que el dispositivo se conecte al menos 12 veces en total. Las marcas de tiempo recopiladas se utilizan para validar que el dispositivo sometido a prueba utiliza el retroceso de fluctuación. Si el dispositivo sometido a prueba tiene un retraso de retroceso estrictamente exponencial, este caso de prueba pasará con advertencias.

Recomendamos la implementación del[Retroceso exponencial y fluctuación](#)mechanismo en el dispositivo que se está probando para pasar este caso de prueba.

Definición de caso de prueba API:

Note

`EXECUTION_TIMEOUT`tiene un valor predeterminado de 5 minutos. Recomendamos un valor de tiempo de espera 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"
    }
  }
]
```

«Reintentos de conexión de dispositivos con retroceso exponencial - Sin respuesta CONNACK»

Valida que el dispositivo sometido a prueba utilice el retroceso exponencial adecuado al volver a conectarse con el agente durante al menos cinco veces. El agente registra la marca de hora del dispositivo bajo la solicitud CONNECT de la prueba, realiza la validación de paquetes, hace una pausa sin enviar un CONNACK al dispositivo cliente y espera a que el dispositivo sometido a prueba vuelva a enviar la solicitud. Las marcas de tiempo recopiladas se utilizan para validar que el dispositivo sometido a prueba utiliza un retroceso exponencial.

Recomendamos la implementación del[Retroceso exponencial y fluctuación](#)mechanismo en el dispositivo que se está probando para pasar este caso de prueba.

Definición de caso de prueba API:

Note

`EXECUTION_TIMEOUT`tiene un valor predeterminado de 5 minutos. Recomendamos un valor de tiempo de espera 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"
}
}]
```

«Vuelva a conectar el dispositivo con retroceso de fluctuación: después de desconectar el servidor»

Valida si un dispositivo sometido a prueba utiliza la fluctuación y el retroceso necesarios mientras se vuelve a conectar después de desconectar del servidor. Device Advisor desconecta el dispositivo del servidor durante al menos cinco veces y observa el comportamiento del dispositivo para la reconexión de MQTT. Device Advisor registra la marca de hora de la solicitud CONNECT del dispositivo que se está probando, realiza la validación de paquetes, hace una pausa sin enviar un CONNACK al dispositivo cliente y espera a que el dispositivo sometido a prueba vuelva a enviar la solicitud. Las marcas de tiempo recopiladas se utilizan para validar que el dispositivo en prueba utiliza fluctuación y retroceso mientras se vuelve a conectar. Si el dispositivo sometido a prueba tiene un retroceso estrictamente exponencial o no implementa un mecanismo de retroceso de jitter adecuado, este caso de prueba pasará con advertencias. Si el dispositivo sometido a prueba ha implementado un mecanismo de retroceso lineal o un mecanismo de retroceso constante, la prueba fallará.

Para aprobar este caso de prueba, recomendamos implementar el[Retroceso exponencial y fluctuación](#)mechanismo del dispositivo bajo prueba.

Definición de caso de prueba API:

Note

`EXECUTION_TIMEOUT`tiene un valor predeterminado de 5 minutos. Recomendamos un valor de tiempo de espera 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"
    }
}]
```

El número de intentos de reconexión que se van a validar para el retroceso se puede cambiar especificando la`RECONNECTION_ATTEMPTS`. El número debe estar comprendido entre cinco y diez. El valor predeterminado es cinco.

Keep-Alive

«Matt No Ak Pinesp»

Este caso de prueba valida si el dispositivo sometido a prueba se desconecta cuando no recibe una respuesta de ping. Como parte de este caso de prueba, Device Advisor bloquea las respuestas

enviadas desde AWS IoT Core para solicitudes de publicación, suscripción y ping. También valida si el dispositivo sometido a prueba desconecta la conexión MQTT.

Definición de caso de prueba API:

Note

`EXECUTION_TIMEOUT` tiene un valor predeterminado de 5 minutos. Recomendamos un tiempo de espera superior a 1,5 veces el `keepAliveTime` valor.

```
"tests": [
  {
    "name": "Mqtt_No_Ack_PingResp",
    "configuration":
      //optional:
      "EXECUTION_TIMEOUT": "306",    // in seconds
    },
    "test": {
      "id": "MQTT_No_Ack_PingResp",
      "version": "0.0.0"
    }
  }
]
```

Publicación

«QoS0 (caso feliz)»

Valida que el dispositivo sometido a prueba publica un mensaje con QoS0. También puede validar el tema del mensaje especificando este valor de tema en la configuración de la prueba.

Definición de caso de prueba API:

Note

`EXECUTION_TIMEOUT` tiene un valor predeterminado de 5 minutos. Recomendamos un valor de tiempo de espera 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"
    }
  }
]
```

«Reintento de publicación QoS1 - Sin PUBACK»

Valida que el dispositivo sometido a prueba vuelve a publicar un mensaje enviado con QoS1, si el agente no envía PUBACK. También puede validar el tema del mensaje especificando este tema en la configuración de la prueba. El dispositivo cliente no debe desconectarse antes de volver a publicar el mensaje. Esta prueba también valida que el mensaje publicado de nuevo tenga el mismo identificador de paquete que el original.

Definición de caso de prueba API:

Note

`EXECUTION_TIMEOUT`tiene un valor predeterminado de 5 minutos. Se recomienda durante al 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"
    }
  }
]
```

Suscribirse

«Puede suscribirse (Happy Case)»

Valida que el dispositivo sometido a prueba se suscriba a temas de MQTT. También puede validar el tema al que se suscribe el dispositivo sometido a prueba especificando este tema en la configuración de la prueba.

Definición de caso de prueba API:

Note

`EXECUTION_TIMEOUT`tiene un valor predeterminado de 5 minutos. Recomendamos un valor de tiempo de espera 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"
      }
    }
]
```

«Reintento de suscripción - Sin SUBACK»

Valida que el dispositivo sometido a prueba vuelve a intentar una suscripción fallida a temas de MQTT. El servidor espera y no envía un SUBACK. Si el dispositivo cliente no vuelve a intentar la suscripción, la prueba falla. El dispositivo cliente debe volver a intentar la suscripción fallida con el mismo ID de paquete. También puede validar el tema al que se suscribe el dispositivo sometido a prueba especificando este tema en la configuración de la prueba.

Definición de caso de prueba API:

Note

`EXECUTION_TIMEOUT`tiene un valor predeterminado de 5 minutos. Recomendamos un valor de tiempo de espera 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"
    }
  }
]
```

Sesión persistente

«Sesión persistente (caso feliz)»

Este caso de prueba valida el comportamiento del dispositivo cuando se desconecta de una sesión persistente. El caso de prueba comprueba si el dispositivo puede volver a conectarse, reanudar las suscripciones a sus temas desencadenadores sin volver a suscribirse explícitamente, recibir los mensajes almacenados en los temas y funciona según lo esperado durante una sesión persistente. Cuando se supera este caso de prueba, indica que el dispositivo cliente puede mantener una sesión persistente con elAWS IoT Corecorredor de la manera esperada. Para obtener más información sobre lasAWS IoT Sesiones persistentes, consulte[Uso de sesiones persistentes de MQTT](#).

En este caso de prueba, se espera que el dispositivo cliente se CONECTE con elAWS IoT Corecon una marca de sesión limpia establecida en false y, a continuación, suscribirse a un tema desencadenador. Tras una suscripción satisfactoria, el dispositivo se desconectará medianteAWS IoT CoreDevice Advisor. Mientras el dispositivo está desconectado, se almacenará una carga útil de mensajes QoS 1 en ese tema. A continuación, Device Advisor permitirá que el dispositivo cliente se vuelva a conectar con el punto final de prueba. En este punto, dado que hay una sesión persistente, se espera que el dispositivo cliente reanude sus suscripciones al tema sin enviar ningún paquete SUBSCRIBE adicional y reciba el mensaje QoS 1 del agente. Después de volver a conectarse, si el dispositivo cliente vuelve a suscribirse a su tema desencadenador enviando un paquete SUBSCRIBE adicional y/o si el cliente no recibe el mensaje almacenado del tema desencadenador, el caso de prueba fallará.

Definición de caso de prueba API:

Note

`EXECUTION_TIMEOUT`tiene un valor predeterminado de 5 minutos. Recomendamos un valor de tiempo de espera de al menos 4 minutos. En la primera conexión, el dispositivo cliente debe suscribirse explícitamente a un`TRIGGER_TOPIC`que no se había suscrito antes. Para aprobar el caso de prueba, el dispositivo cliente debe suscribirse correctamente a`TRIGGER_TOPIC`con QoS 1. Después de volver a conectarse, se espera que el dispositivo cliente comprenda que hay una sesión persistente activa; por lo tanto, debería aceptar el mensaje almacenado enviado por el tema de activación y devolver PUBACK para ese mensaje específico.

```
"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"
    }
  }
]
```

«Sesión persistente: caducidad de la sesión»

Este caso de prueba ayuda a validar el comportamiento del dispositivo cuando un dispositivo desconectado se vuelve a conectar a una sesión persistente caducada. Una vez que finalice la sesión, esperamos que el dispositivo se vuelva a suscribir a los temas a los que se suscribió previamente enviando explícitamente un nuevo paquete SUBSCRIBE.

Durante la primera conexión, esperamos que el dispositivo de prueba se CONECTE con el AWSBroker de IoT, como CleanSession se establece en false para iniciar una sesión persistente. A continuación, el dispositivo debe suscribirse a un tema desencadenador. A continuación, el dispositivo se desconecta mediante AWS IoT CoreDevice Advisor, tras una suscripción satisfactoria e iniciación de una sesión persistente. Tras la desconexión, AWS IoT CoreDevice Advisor permite que el dispositivo de prueba vuelva a conectarse con el punto final de prueba. En este punto, cuando el dispositivo de prueba envía otro paquete CONNECT, AWS IoT CoreDevice Advisor devuelve un paquete CONNACK que indica que la sesión persistente ha caducado. El dispositivo de prueba debe interpretar correctamente este paquete y se espera que vuelva a suscribirse al mismo tema desencadenador cuando finaliza la sesión persistente. Si el dispositivo de prueba no vuelve a suscribirse a su desencadenador de tema, el caso de prueba falla. Para que la prueba pase, el dispositivo debe comprender que la sesión persistente ha terminado y enviar un nuevo paquete SUBSCRIBE para el mismo tema de activación en la segunda conexión.

Si este caso de prueba pasa para un dispositivo de prueba, indica que el dispositivo puede gestionar la reconexión al vencer la sesión persistente de una forma esperada.

Definición de caso de prueba API:

Note

EXECUTION_TIMEOUT tiene un valor predeterminado de 5 minutos. Recomendamos un valor de tiempo de espera de al menos 4 minutos. El dispositivo de prueba debe suscribirse explícitamente a un TRIGGER_TOPIC, al que no se había suscrito antes. Para aprobar el caso de prueba, el dispositivo de prueba debe enviar un paquete CONNECT con CleanSession marca establecida en false y se suscribe correctamente a un tema desencadenador con QoS 1. Tras una conexión satisfactoria, AWS IoT CoreDevice Advisor desconecta el dispositivo. Tras la desconexión, AWS IoT CoreDevice Advisor permite que el dispositivo vuelva a conectarse y se espera que el dispositivo vuelva a suscribirse al mismo TRIGGER_TOPIC since AWS IoT CoreDevice Advisor habría finalizado la sesión 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"
}
]
```

Sombra

Utilice estas pruebas para verificar los dispositivos en uso de prueba AWS IoT Servicio Device Shadow de correctamente. Para obtener más información, consulte [Servicio Device Shadow de AWS IoT \(p. 627\)](#). Si estos casos de prueba están configurados en el conjunto de pruebas, es necesario proporcionar algo al iniciar la ejecución de la suite.

Publicación

«El dispositivo publica el estado después de que se conecta (caso feliz)»

Valida si un dispositivo puede publicar su estado después de conectarse a AWS IoT Core

Definición de caso de prueba API:

Note

`EXECUTION_TIMEOUT`tiene un valor predeterminado de 5 minutos. Recomendamos un valor de tiempo de espera 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"
    }
}
]
```

La `REPORTED_STATE` se puede proporcionar para una validación adicional del estado de sombra exacto del dispositivo después de conectarse. De forma predeterminada, este caso de prueba valida el estado de publicación del dispositivo.

Si `SHADOW_NAME` no se proporciona, el caso de prueba busca los mensajes publicados en los prefijos de tema del tipo de sombra sin nombre (clásico) de forma predeterminada. Proporcione un nombre de sombra si el dispositivo utiliza el tipo de sombra con nombre. Consulte [Uso de sombras en dispositivos](#) para obtener más información.

Actualización

«Actualizaciones del dispositivo notificadas al estado deseado (caso feliz)»

Valida si el dispositivo lee todos los mensajes de actualización recibidos y sincroniza el estado del dispositivo para que coincida con las propiedades de estado deseadas. El dispositivo debe publicar su último estado informado tras la sincronización. Si el dispositivo ya tiene una sombra existente antes de ejecutar la prueba, asegúrese de que el estado deseado configurado para el caso de prueba y el estado notificado existente no coinciden ya. Puede identificar los mensajes de actualización de sombra enviados por Device Advisor consultando elClientTokenen el documento Shadow tal como `seráDeviceAdvisorShadowTestCaseSetup`.

Definición de caso de prueba API:

Note

`EXECUTION_TIMEOUT`tiene un valor predeterminado de 5 minutos. Recomendamos un valor de tiempo de espera 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"
    }
  }
]
```

La`DESIRED_STATE`debe tener al menos un atributo y un valor asociado.

Si`SHADOW_NAME`no se proporciona, a continuación, el caso de prueba busca los mensajes publicados en los prefijos de tema del tipo de sombra sin nombre (clásico) de forma predeterminada. Proporcione un nombre de sombra si el dispositivo utiliza el tipo de sombra con nombre. Consulte[Uso de sombras en dispositivos](#)para obtener más información.

Ejecución de Job

«El dispositivo puede completar la ejecución de un trabajo»

Este caso de prueba te ayuda a validar si tu dispositivo puede recibir actualizaciones medianteAWS IoTTrabajos y publica el estado de las actualizaciones correctas. Para obtener más información sobre lasAWS IoTTrabajos, consulte[Trabajos](#).

Definición de caso de prueba API:

Note

`EXECUTION_TIMEOUT`tiene un valor predeterminado de 5 minutos. Recomendamos un valor de tiempo de espera 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": "{\n        \"operation\": \"reboot\",\n        \"files\": {\n            \"fileName\": \"install.py\",\n            \"url\": \"${aws:iot:s3-presigned-url:https://s3.amazonaws.com/bucket-\nname/key}\\\"\n        }\n    }",
      // 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 obtener más información acerca de la creación y uso de documentos de trabajo,
consulte[documento de trabajo](#).

Mensajes de los eventos

Esta sección contiene información sobre los mensajes publicados por AWS IoT cuando se actualizan o se modifican objetos o trabajos. Para obtener información acerca de la AWS IoT Eventsservicio que le permite crear detectores para monitorizar los dispositivos con la finalidad de ver si se producen errores o cambios en su funcionamiento, y para activar acciones específicas cuando se produzcan, consulte [AWS IoT Events](#).

Cómo se generan los mensajes de eventos

AWS IoT publica mensajes de eventos cuando se producen determinados eventos. Por ejemplo, el registro genera eventos cuando se añaden, actualizan o eliminan objetos. Cada evento provoca que se envíe un único mensaje de evento. Los mensajes de evento se publican a través de MQTT con una carga JSON. El contenido de la carga depende del tipo de evento.

Note

Se garantiza que los mensajes de eventos se publican una vez. Es posible que se publiquen más de una. No se garantiza el orden de los mensajes de eventos.

Política de recepción de mensajes de eventos

Para recibir mensajes de eventos, el dispositivo debe usar una política adecuada que le permita conectarse a la gateway de dispositivos de AWS IoT y suscribirse a los temas de eventos de MQTT. También debe suscribirse a los filtros de temas adecuados.

A continuación, mostramos un ejemplo de política necesaria para recibir eventos del ciclo de vida:

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "iot:Subscribe",  
                "iot:Receive"  
            ],  
            "Resource": [  
                "arn:aws:iot:region:account:/$aws/events/*"  
            ]  
        }  
    ]  
}
```

Habilitar eventos para AWS IoT

Antes de que los suscriptores a los temas reservados puedan recibir mensajes, debe habilitar los mensajes de eventos desde el AWS Management Consoleo mediante la API o la CLI. Para obtener información acerca de los mensajes de eventos que administran las distintas opciones de, consulte la [Tabla de AWS IoT configuración de eventos \(p. 1098\)](#).

- Para habilitar los mensajes de eventos, vaya a la sección [Configuración de AWS IoT](#) consola y, a continuación, en el Mensajes basados en eventos sección, elija Administración de eventos. Puede especificar los eventos que desea administrar.
- Para controlar qué tipos de eventos se publican mediante la API o la CLI, llame al [UpdateEventConfigurations](#) API o utilice el `update-event-configurations` Command de la CLI. Por ejemplo:

```
aws iot update-event-configurations --event-configurations "{\"THING\":{\"Enabled\":true}}"
```

Note

Todas las comillas ("") van precedidas de barras diagonales invertidas ()).

Puede consultar las configuraciones de eventos actuales mediante una llamada a la API [DescribeEventConfigurations](#) o utilizando el comando de la CLI describe-event-configurations. Por ejemplo: .

```
aws iot describe-event-configurations
```

Tabla deAWS IoTconfiguración de eventos

Categoría de eventos (AWS IoTConsola: Configuración: mensajes basados en eventos)	eventConfigurations Valor de la clave (AWS CLI/API)	Tema de mensaje de evento
(Solo se puede configurar mediante elAWS CLI/API)	CA_CERTIFICATE	\$aws/events/ certificates/ registered/ <i>caCertificateId</i>
(Solo se puede configurar mediante elAWS CLI/API)	CERTIFICATE	\$aws/events/presence/ connected/ <i>clientId</i>
(Solo se puede configurar mediante elAWS CLI/API)	CERTIFICATE	\$aws/events/presence/ disconnected/ <i>clientId</i>
(Solo se puede configurar mediante elAWS CLI/API)	CERTIFICATE	\$aws/events/ subscriptions/ subscribed/ <i>clientId</i>
(Solo se puede configurar mediante elAWS CLI/API)	CERTIFICATE	\$aws/events/ subscriptions/ unsubscribed/ <i>clientId</i>
Job completado, cancelado	JOB	\$aws/events/job/ <i>jobID</i> / canceled
Job completado, cancelado	JOB	\$aws/events/job/ <i>jobID</i> / cancellation_in_progress
Job completado, cancelado	JOB	\$aws/events/job/ <i>jobID</i> / completed
Job completado, cancelado	JOB	\$aws/events/job/ <i>jobID</i> / deleted
Job completado, cancelado	JOB	\$aws/events/job/ <i>jobID</i> / deletion_in_progress

Categoría de eventos (AWS IoTConsola: Configuración: mensajes basados en eventos)	eventConfigurations Valor de la clave (AWS CLI/API)	Tema de mensaje de evento
Ejecución del Job: correcta, fallida, rechazada, cancelada, eliminada	JOB_EXECUTION	\$aws/events/ jobExecution/ <i>jobID</i> / canceled
Ejecución del Job: correcta, fallida, rechazada, cancelada, eliminada	JOB_EXECUTION	\$aws/events/ jobExecution/ <i>jobID</i> / deleted
Ejecución del Job: correcta, fallida, rechazada, cancelada, eliminada	JOB_EXECUTION	\$aws/events/ jobExecution/ <i>jobID</i> / failed
Ejecución del Job: correcta, fallida, rechazada, cancelada, eliminada	JOB_EXECUTION	\$aws/events/ jobExecution/ <i>jobID</i> / rejected
Ejecución del Job: correcta, fallida, rechazada, cancelada, eliminada	JOB_EXECUTION	\$aws/events/ jobExecution/ <i>jobID</i> / removed
Ejecución del Job: correcta, fallida, rechazada, cancelada, eliminada	JOB_EXECUTION	\$aws/events/ jobExecution/ <i>jobID</i> / succeeded
Ejecución del Job: correcta, fallida, rechazada, cancelada, eliminada	JOB_EXECUTION	\$aws/events/ jobExecution/ <i>jobID</i> / timed_out
Cosa: creada, actualizada, eliminada	THING	\$aws/events/ thing/ <i>thingName</i> /created
Cosa: creada, actualizada, eliminada	THING	\$aws/events/ thing/ <i>thingName</i> /updated
Cosa: creada, actualizada, eliminada	THING	\$aws/events/ thing/ <i>thingName</i> /deleted
Grupo de cosas: agregado, eliminado	THING_GROUP	\$aws/events/ thingGroup/ <i>thingGroupName</i> /created
Grupo de cosas: agregado, eliminado	THING_GROUP	\$aws/events/ thingGroup/ <i>thingGroupName</i> /updated
Grupo de cosas: agregado, eliminado	THING_GROUP	\$aws/events/ thingGroup/ <i>thingGroupName</i> /deleted

Categoría de eventos (AWS IoTConsola: Configuración: mensajes basados en eventos)	eventConfigurations Valor de la clave (AWS CLI/API)	Tema de mensaje de evento
Jerarquía de grupos de cosas: agregada, eliminada	THING_GROUP_HIERARCHY	\$aws/events/ thingGroupHierarchy/ thingGroup/ <i>parentThingGroupName</i> / childThingGroup/ <i>childThingGroupName</i> / added
Jerarquía de grupos de cosas: agregada, eliminada	THING_GROUP_HIERARCHY	\$aws/events/ thingGroupHierarchy/ thingGroup/ <i>parentThingGroupName</i> / childThingGroup/ <i>childThingGroupName</i> / removed
Pertenencia a un grupo de cosas: agregada,	THING_GROUP_MEMBERSHIP	\$aws/events/ thingGroupMembership/ thingGroup/ <i>thingGroupName</i> / thing/ <i>thingName</i> /added
Pertenencia a un grupo de cosas: agregada,	THING_GROUP_MEMBERSHIP	\$aws/events/ thingGroupMembership/ thingGroup/ <i>thingGroupName</i> / thing/ <i>thingName</i> /removed
Tipo de cosa: creado, actualizado, eliminado	THING_TYPE	\$aws/events/ thingType/ <i>thingTypeName</i> / created
Tipo de cosa: creado, actualizado, eliminado	THING_TYPE	\$aws/events/ thingType/ <i>thingTypeName</i> / updated
Tipo de cosa: creado, actualizado, eliminado	THING_TYPE	\$aws/events/ thingType/ <i>thingTypeName</i> / deleted
Asociación de tipo de cosa: agregada, eliminada	THING_TYPE_ASSOCIATION	\$aws/events/ thingTypeAssociation/ thing/ <i>thingName</i> / <i>thingTypeName</i>

Eventos de registro

El registro puede publicar mensajes de eventos cuando se crean, actualizan o eliminan objetos, tipos de objetos y grupos de objetos. Sin embargo, estos eventos no están disponibles de forma predeterminada. Para obtener información acerca de cómo activar estos eventos, consulte [Habilitar eventos para AWS IoT \(p. 1097\)](#).

El registro puede proporcionar los siguientes tipos de eventos:

- [Eventos de objeto \(p. 1101\)](#)
- [Eventos de tipo de objeto \(p. 1102\)](#)
- [Eventos de grupos de objetos \(p. 1104\)](#)

Eventos de objeto

Objeto creado, actualizado o eliminado

El registro publica los siguientes mensajes de eventos cuando se crean, actualizan o eliminan objetos:

- `$aws/events/thing/thingName/created`
- `$aws/events/thing/thingName/updated`
- `$aws/events/thing/thingName/deleted`

Los mensajes contienen la siguiente carga de ejemplo:

```
{  
    "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"  
    }  
}
```

Las cargas contienen los siguientes atributos:

eventType

Se establece en "THING_EVENT".

eventId

Un ID de evento exclusivo (cadena).

timestamp

La marca de tiempo UNIX de cuándo se produjo el evento.

operación

La operación en la que se activó el evento. Los valores válidos son:

- CREATED
- UPDATED
- DELETED

accountId

SusCuenta de AWSID.

thingId

El ID del objeto que se crea, actualiza o elimina.

thingName

El nombre del objeto que se crea, actualiza o elimina.

versionNumber

La versión del objeto que se crea, actualiza o elimina. Este valor se establece en 1 cuando se crea un objeto. Aumenta en 1 cada vez que se actualiza el objeto.

thingTypeName

El tipo de objeto asociado al objeto, si existiera. De lo contrario, null.

attributes

Un conjunto de pares nombre-valor asociados al objeto.

Eventos de tipo de objeto

Eventos relacionados con el tipo de cosa:

- [Tipo de objeto creado, descartado, eliminado o al que se le ha quitado la marca de descartado \(p. 1102\)](#)
- [Tipo de objeto asociado o desasociado de un objeto \(p. 1103\)](#)

Tipo de objeto creado, descartado, eliminado o al que se le ha quitado la marca de descartado

El registro publica los siguientes mensajes de eventos cuando se crean, se descartan, se eliminan o se quita la marca de descartado de los tipos de objetos:

- \$aws/events/thingType/*thingTypeName*/created
- \$aws/events/thingType/*thingTypeName*/updated
- \$aws/events/thingType/*thingTypeName*/deleted

El mensaje contiene la siguiente carga de ejemplo:

```
{  
    "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"  
}
```

Las cargas contienen los siguientes atributos:

eventType

Se establece en "THING_TYPE_EVENT".

eventId

Un ID de evento exclusivo (cadena).

timestamp

La marca de tiempo UNIX de cuándo se produjo el evento.

operación

La operación en la que se activó el evento. Los valores válidos son:

- CREATED
- UPDATED
- DELETED

accountId

SusCuenta de AWSID.

thingTypeId

El ID del tipo de objeto que se crea o elimina, o que está descartado.

thingTypeName

El nombre del tipo de objeto que se crea o elimina, o que está descartado.

isDeprecated

`true` si el tipo de objeto está descartado. De lo contrario, `false`.

deprecationDate

La marca de tiempo UNIX para cuando el tipo de objeto está descartado.

searchableAttributes

Un conjunto de pares nombre-valor asociados con el tipo de objeto que puede utilizarse para realizar búsquedas.

description

Una descripción del tipo de objeto.

Tipo de objeto asociado o desasociado de un objeto

El registro publica los siguientes mensajes de eventos cuando se asocia o desasocia un tipo de objeto a un objeto.

- `$aws/events/thingTypeAssociation/thing/thingName/typeName`

Los mensajes contienen la siguiente carga de ejemplo:

```
{  
    "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,  
}
```

Las cargas contienen los siguientes atributos:

eventId

Un ID de evento exclusivo (cadena).

eventType

Se establece en "THING_TYPE_ASSOCIATION_EVENT".

operación

La operación en la que se activó el evento. Los valores válidos son:

- CREATED
- DELETED

thingId

El ID del objeto cuya asociación de tipo ha cambiado.

thingName

El nombre del objeto cuya asociación de tipo ha cambiado.

thingTypeName

El tipo de objeto asociado o desasociado del objeto.

timestamp

La marca de tiempo UNIX de cuándo se produjo el evento.

Eventos de grupos de objetos

Eventos relacionados con grupos de cosas:

- Grupo de objetos creado, actualizado o eliminado (p. 1104)
- Objeto añadido o eliminado de un grupo de objetos (p. 1106)
- Grupo de objetos añadido o eliminado de un grupo de objetos (p. 1107)

Grupo de objetos creado, actualizado o eliminado

El registro publica los siguientes mensajes de eventos cuando se crea, actualiza o elimina un grupo de objetos.

- \$aws/events/thingGroup/*groupName*/created
- \$aws/events/thingGroup/*groupName*/updated
- \$aws/events/thingGroup/*groupName*/deleted

A continuación se muestra un ejemplo deupdatedcarga. Cargas útiles paracreatedydeletedlos mensajes son similares.

```
{  
    "eventType": "THING_GROUP_EVENT",  
    "eventId": "8b9ea8626aea1e42100f3f32b975899",  
    "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"  

```

Las cargas contienen los siguientes atributos:

eventType

Se establece en "THING_GROUP_EVENT".

eventId

Un ID de evento exclusivo (cadena).

timestamp

La marca de tiempo UNIX de cuándo se produjo el evento.

operación

La operación en la que se activó el evento. Los valores válidos son:

- CREATED
- UPDATED
- DELETED

accountId

SusCuenta de AWSID.

thingGroupId

El ID del grupo de objetos que se crea, actualiza o elimina.

thingGroupName

El nombre del grupo de objetos que se crea, actualiza o elimina.

versionNumber

La versión del grupo de objetos. Este valor se establece en 1 cuando se crea un grupo de objetos.
Aumenta en 1 cada vez que se actualiza el grupo de objetos.

parentGroupName

El nombre del grupo principal de objetos, si existe.

parentGroupId

El ID del grupo principal de objetos, si existe.

description

Una descripción del grupo de objetos.

rootToParentThingGroups

Una matriz de información acerca del grupo principal de objetos. Hay un elemento para cada grupo de objetos matriz, que empieza por el grupo de objetos raíz y continúa hasta el elemento principal del grupo de objetos. Cada entrada contiene el grupo de objetos `groupArn` y `groupId`.

attributes

Un conjunto de pares nombre-valor asociados al grupo de objetos.

Objeto añadido o eliminado de un grupo de objetos

El registro publica los siguientes mensajes de eventos cuando se añade un objeto a un grupo de objetos o se elimina de este.

- `$aws/events/thingGroupMembership/thingGroup/thingGroupName/thing/thingName/added`
- `$aws/events/thingGroupMembership/thingGroup/thingGroupName/thing/thingName/removed`

Los mensajes contienen la siguiente carga de ejemplo:

```
{  
    "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"  
}
```

Las cargas contienen los siguientes atributos:

eventType

Se establece en "THING_GROUP_MEMBERSHIP_EVENT".

eventId

El ID del evento.

timestamp

La marca de tiempo UNIX de cuándo se produjo el evento.

operación

ADDED cuando se añade un objeto a un grupo de objetos. REMOVED cuando se elimina un objeto de un grupo de objetos.

accountId

SusCuenta de AWSID.

groupArn

El ARN del grupo de objetos.

groupId

El ID del grupo.

thingArn

El ARN del objeto que se añadió o quitó del grupo de objetos.

thingId

El ID del objeto que se añadió o quitó del grupo de objetos.

membershipId

Un ID que representa la relación entre objeto y el grupo de objetos. Este valor se genera cuando añade un objeto a un grupo de objetos.

Grupo de objetos añadido o eliminado de un grupo de objetos

El registro publica los siguientes mensajes de eventos cuando se añade un grupo de objetos a otro grupo de objetos o se elimina de este.

- \$aws/events/thingGroupHierarchy/thingGroup/*parentThingGroupName*/childThingGroup/*childThingGroupName*/added
- \$aws/events/thingGroupHierarchy/thingGroup/*parentThingGroupName*/childThingGroup/*childThingGroupName*/removed

El mensaje contiene la siguiente carga de ejemplo:

```
{  
    "eventType" : "THING_GROUP_HIERARCHY_EVENT",  
    "eventId" : "264192c7-b573-46ef-ab7b-489fc4d47da41",  
    "timestamp" : 1234567890123,  
    "operation" : "ADDED|REMOVED",  
    "accountId" : "123456789012",  
    "thingGroupId" : "8f82a106-6b1d-4331-8984-a84db5f6f8cb",  
    "thingGroupName" : "MyRootThingGroup",  
    "childGroupId" : "06838589-373f-4312-b1f2-53f2192291c4",  
    "childGroupName" : "MyChildThingGroup"  
}
```

Las cargas contienen los siguientes atributos:

eventType

Se establece en "THING_GROUP_HIERARCHY_EVENT".

eventId

El ID del evento.

timestamp

La marca de tiempo UNIX de cuándo se produjo el evento.

operación

ADDED cuando se añade un objeto a un grupo de objetos. REMOVED cuando se elimina un objeto de un grupo de objetos.

accountId

SusCuenta de AWSID.

thingGroupId

El ID del grupo principal de objetos.

thingGroupName

El nombre del grupo principal de objetos.

childGroupId

El ID del grupo secundario de objetos.

childGroupName

El nombre del grupo secundario de objetos.

Eventos de trabajos

La AWS IoT servicio Jobs de publica en temas reservados en el protocolo MQTT cuando los trabajos están pendientes, completados o cancelados y cuando un dispositivo informa del éxito o del fracaso cuando se ejecuta un trabajo. Los dispositivos o las aplicaciones de administración y monitorización pueden realizar un seguimiento del estado de los trabajos mediante la suscripción a estos temas.

Cómo habilitar los eventos de trabajo

Mensajes de respuesta del AWS IoT servicio de trabajos no pasa por el agente de mensajes y otros clientes o reglas no pueden suscribirse a ellos. Para suscribirse a mensajes relacionados con la actividad de trabajo, utilice el `notify`-`next` temas. Para obtener información sobre temas de trabajos, consulte [Temas de trabajos \(p. 107\)](#).

Para recibir notificaciones de actualizaciones de trabajos, habilite estos eventos de trabajos mediante el AWS Management Console, o mediante la API o la CLI de. Para obtener más información, consulte [Habilitar eventos para AWS IoT \(p. 1097\)](#).

Cómo funcionan los eventos de empleo

Dado que cancelar o eliminar un trabajo puede llevar un tiempo, se envían dos mensajes para indicar el comienzo y el final de una solicitud. Por ejemplo, cuando se inicia una solicitud de cancelación, se envía un mensaje al tema `$aws/events/job/jobID/cancellation_in_progress`. Cuando finaliza una solicitud de cancelación, se envía un mensaje al tema `$aws/events/job/jobID/canceled`.

En las solicitudes de eliminación de trabajos se lleva a cabo un proceso parecido. Las aplicaciones de administración y monitorización pueden suscribirse a estos temas y hacer un seguimiento del estado de los trabajos. Para obtener más información acerca de cómo publicar en temas de MQTT y suscribirse a ellos, consulte [the section called “Protocolos de comunicación de dispositivos” \(p. 81\)](#).

Tipos de eventos de Job

A continuación se muestran los distintos tipos de eventos de trabajo:

Trabajo completado, cancelado o eliminado

El servicio Jobs de AWS IoT publica un mensaje en un tema de MQTT cuando se completa, cancela o elimina un trabajo, o bien cuando la cancelación o la eliminación está en curso:

- \$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

El mensaje completed contiene la siguiente carga de ejemplo:

```
{  
  "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  
  }  
}
```

La canceled contiene la siguiente carga de ejemplo.

```
{  
  "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
}
```

La `deleted` contiene la siguiente carga de ejemplo.

```
{
  "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"
}
```

El mensaje `cancellation_in_progress` contiene la siguiente carga de ejemplo:

```
{
  "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"
}
```

El mensaje `deletion_in_progress` contiene la siguiente carga de ejemplo:

```
{
  "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"
}
```

Estado final de ejecución de trabajo

El servicio Jobs de AWS IoT publica un mensaje cuando un dispositivo actualiza la ejecución de un trabajo al estado final:

- \$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

El mensaje contiene la siguiente carga de ejemplo:

```
{
  "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 del ciclo de vida

AWS IoT puede publicar eventos del ciclo de vida en los temas MQTT. Estos eventos están disponibles de forma predeterminada y no se pueden deshabilitar.

Note

Es posible que los mensajes de ciclo de vida se envíen de forma desordenada. Puede que reciba mensajes duplicados.

Eventos de conexión/desconexión

AWS IoT publica un mensaje en los siguientes temas MQTT cuando un cliente se conecta o se desconecta:

- \$aws/events/presence/connected/*clientId*: un cliente se ha conectado al agente de mensajes.
- \$aws/events/presence/disconnected/*clientId*— Un cliente se ha desconectado del agente de mensajes.

A continuación, se muestra una lista de elementos JSON que se encuentran en los mensajes de conexión/desconexión publicados en el tema `$aws/events/presence/connected/clientId`.

clientId

El ID del cliente que se conecta o se desconecta.

Note

Los ID de cliente que contienen los símbolos # o + no recibirán eventos del ciclo de vida.

clientInitiatedDisconnect

True si el cliente inició la desconexión. De lo contrario, devuelve false. Solo se encuentra en los mensajes de desconexión.

disconnectReason

La razón por la que el cliente se está desconectando. Sólo se encuentra en mensajes de desconexión. La tabla siguiente contiene valores válidos.

Motivo de desconexión	Descripción
AUTH_ERROR	El cliente no pudo autenticarse o la autorización devolvió un error.
CLIENT_INITIATED_DISCONNECT	El cliente indica que se desconectará. El cliente puede hacer esto enviando un MQTT DISCONNECT paquete de control o un close framesi el cliente utiliza un WebSocket conexión de.
CLIENT_ERROR	El cliente hizo algo mal que provocó su desconexión. Por ejemplo, un cliente se desconectará por enviar más de un paquete CONNECT de MQTT en la misma conexión o si el cliente intenta publicar con una carga útil que supera el límite de carga útil.
CONNECTION_LOST	La conexión cliente-servidor está cortada. Esto puede ocurrir durante un período de alta latencia de red o cuando se pierde la conexión a Internet.
DUPLICATE_CLIENTID	El cliente está utilizando un ID de cliente que ya está en uso. En este caso, el cliente que ya está conectado se desconectará con esta razón de desconexión.
FORBIDDEN_ACCESS	No se permite la conexión del cliente. Por ejemplo, un cliente con una dirección IP denegada no podrá conectarse.
MQTT_KEEP_ALIVE_TIMEOUT	Si no hay comunicación cliente-servidor para 1,5 veces el tiempo de mantenimiento del cliente, el cliente se desconecta.
SERVER_ERROR	Desconectado debido a problemas inesperados del servidor.
SERVER_INITIATED_DISCONNECT	El servidor desconecta de forma intencionada un cliente por razones operativas.

Motivo de desconexión	Descripción
THROTTLED	El cliente se desconecta por exceder una limitación controlada.
WEBSOCKET_TTL_EXPIRATION	El cliente está desconectado porque un WebSocket se ha conectado un tiempo superior a su valor de tiempo de vida.

eventType

El tipo de evento. Los valores válidos son `connected` o `disconnected`.

ipAddress

La dirección IP del cliente que se conecta. Puede estar en formato IPv4 o IPv6. Sólo se encuentra en los mensajes de conexión.

principalIdentifier

Las credenciales que se utilizan para la autenticación. En el caso de los certificados de autenticación mutua de TLS, se trata del ID de certificado. En cuanto a las demás conexiones, se trata de las credenciales de IAM.

sessionIdentifier

Un identificador único global de AWS IoT que existe durante toda la vida de la sesión.

timestamp

Aproximación del momento en que se produjo el evento, expresada en milisegundos según la fecha de inicio Unix. La precisión de la marca de tiempo es de +/- 2 minutos.

versionNumber

El número de versión del evento del ciclo de vida. Se trata de un valor entero largo que aumenta de forma monótona para cada conexión de un ID de cliente. El número de versión puede utilizarlo un suscriptor para deducir el orden de los eventos del ciclo de vida.

Note

Los mensajes de conexión y desconexión de una conexión de cliente tienen el mismo número de versión.

El número de versión podría saltarse algunos valores y no se garantiza que se vaya a incrementar de forma coherente en 1 para cada evento.

Si un cliente no se conecta durante aproximadamente una hora, el número de versión se restablece a 0. Para las sesiones persistentes, el número de versión se restablece a 0 después de que un cliente se haya desconectado durante un periodo mayor que el configurado time-to-live (TTL) para la sesión persistente.

Un mensaje de conexión tiene la siguiente estructura.

```
{
  "clientId": "186b5",
  "timestamp": 1573002230757,
  "eventType": "connected",
  "sessionIdentifier": "a4666d2a7d844ae4ac5d7b38c9cb7967",
  "principalIdentifier": "12345678901234567890123456789012",
  "ipAddress": "192.0.2.0",
  "versionNumber": 0
}
```

Un mensaje de desconexión tiene la siguiente estructura.

```
{  
    "clientId": "186b5",  
    "timestamp": 1573002340451,  
    "eventType": "disconnected",  
    "sessionIdentifier": "a4666d2a7d844ae4ac5d7b38c9cb7967",  
    "principalIdentifier": "12345678901234567890123456789012",  
    "clientInitiatedDisconnect": true,  
    "disconnectReason": "CLIENT_INITIATED_DISCONNECT",  
    "versionNumber": 0  
}
```

Gestión de desconexiones del cliente

La práctica recomendada consiste siempre en tener implementado un estado de espera para los eventos del ciclo de vida, incluidos los mensajes Last Will and Testament (LWT). Cuando se recibe un mensaje de desconexión, el código debe esperar un periodo de tiempo y verificar que un dispositivo sigue sin conexión antes de tomar cualquier medida. Una forma de hacerlo consiste en utilizar [colas con retraso de SQS](#). Cuando un cliente recibe un evento de ciclo de vida o un mensaje LWT, se puede poner en cola un mensaje (por ejemplo, durante 5 segundos). Cuando dicho mensaje está disponible y se procesa (por parte de Lambda u otro servicio), primero se puede comprobar si el dispositivo sigue sin conexión antes de tomar otras medidas.

Eventos de suscripción/cancelación de suscripción

AWS IoT publica un mensaje en el tema MQTT siguiente cuando un cliente se suscribe a un tema MQTT o cancela su suscripción a este:

```
$aws/events/subscriptions/subscribed/clientId
```

o bien

```
$aws/events/subscriptions/unsubscribed/clientId
```

Donde *clientId* es el ID de cliente MQTT que se conecta con el agente de mensajes de AWS IoT.

El mensaje publicado en este tema tiene la estructura siguiente:

```
{  
    "clientId": "186b5",  
    "timestamp": 1460065214626,  
    "eventType": "subscribed" | "unsubscribed",  
    "sessionIdentifier": "00000000-0000-0000-0000-000000000000",  
    "principalIdentifier": "000000000000/ABCDEFGHIJKLMNPQRSTUVWXYZ:some-user/  
ABCDEFGHIJKLMNPQRSTUVWXYZ:some-user",  
    "topics" : ["foo/bar", "device/data", "dog/cat"]  
}
```

A continuación, se ofrece una lista de elementos JSON que se encuentran en los mensajes suscritos y no suscritos publicados en los temas `$aws/events/subscriptions/subscribed/clientId` y `$aws/events/subscriptions/unsubscribed/clientId`.

clientId

El ID del cliente que se suscribe o cancela su suscripción.

Note

Los ID de cliente que contienen los símbolos # o + no recibirán eventos del ciclo de vida.
eventType

El tipo de evento. Los valores válidos son `subscribed` o `unsubscribed`.
principalIdentifier

Las credenciales que se utilizan para la autenticación. En el caso de los certificados de autenticación mutua de TLS, se trata del ID de certificado. En cuanto a las demás conexiones, se trata de las credenciales de IAM.

sessionId

Un identificador único global de AWS IoT que existe durante toda la vida de la sesión.
timestamp

Aproximación del momento en que se produjo el evento, expresada en milisegundos según la fecha de inicio Unix. La precisión de la marca de tiempo es de +/- 2 minutos.

topics

Una matriz de los temas MQTT a los que se ha suscrito el cliente.

Note

Es posible que los mensajes de ciclo de vida se envíen de forma desordenada. Puede que reciba mensajes duplicados.

AWS IoT Corefor LoRaWAN

AWS IoT Corepara LoRaWAN es un servidor de red (LNS) de LoRaWAN totalmente administrado que proporciona administración de puertas de enlace mediante las capacidades Servidor de configuración y actualización (CUPS) y Actualizaciones de firmware por aire (FUOTA). Puede reemplazar su LNS privado porAWS IoT Corepara LoRaWAN y conecte sus dispositivos y puertas de enlace de red de área extendida de largo alcance (LoRaWAN) aAWS IoT Core. Al hacerlo, reducirá el mantenimiento, los costes operativos, el tiempo de configuración y los costes generales.

Introducción

Los dispositivos LoRaWAN son dispositivos de largo alcance, de bajo consumo y que funcionan con baterías que utilizan el protocolo LoRaWAN para operar en un espectro de radio sin licencia. LoRaWAN es un protocolo de comunicación de red de área amplia de bajo consumo (LPWAN) que se basa en LoRa. LoRa es el protocolo de capa física que permite la comunicación de área amplia y de baja potencia entre dispositivos.

Puede incorporar sus dispositivos LoraWAN de la misma manera que incorporaría otros dispositivos IoT aAWS IoT. Para conectar los dispositivos LoraWAN aAWS IoT, debe usar una puerta de enlace LoRaWAN. La puerta de enlace actúa como puente al que conectar el dispositivoAWS IoT Corepara LoRawan e intercambiar mensajes.AWS IoT Corefor LoRaWAN utilizaAWS IoTmotor de reglas para enrutar los mensajes de sus dispositivos LoraWAN a otrosAWS IoTServicios de .

Para reducir el esfuerzo de desarrollo e incorporar rápidamente sus dispositivos aAWS IoT CorePara LoRawan, le recomendamos que utilice dispositivos de enlace con certificación Lorawan. Para obtener más información, consulte la[AWS IoT Corefor LoRaWAN](#)(Se ha creado el certificado). Para obtener información acerca de cómo obtener la certificación de sus dispositivos LoRaWAN de, consulte[Certificación de productos LoRaWAN](#).

Cómo utilizarAWS IoT Corefor LoRaWAN

Puede incorporar rápidamente sus dispositivos LoraWAN y puertas de enlace aAWS IoT Corepara LoRawan mediante la consola o laAWS IoTAPI inalámbrica.

Con la consola

Para incorporar sus dispositivos y puertas de enlace LoraWAN mediante elAWS Management Console, inicie sesión en elAWS Management Consoley vaya a la[AWS IoT Corefor LoRaWAN](#)en laAWS IoTconsola de . A continuación, puede utilizar ellintropara agregar puertas de enlace y dispositivos aAWS IoT Corefor LoRaWAN. Para obtener más información, consulte [Uso de la consola para incorporar el dispositivo y la puerta de enlace aAWS IoT Core for LoRaWAN \(p. 1121\)](#).

Uso de la API o la CLI

Puedes incorporar tanto LoRaWAN comoAceradispositivos mediante el[AWS IoTWireless](#)API. LaAWS IoTAPI inalámbrica queAWS IoT Corepara LoRaWAN sobre el que se basa está respaldado por elAWSSDK. Para obtener más información, consulte[AWSSDK y conjuntos de herramientas](#).

Puede utilizar elAWS CLIpaa ejecutar comandos de incorporación y administración de las puertas de enlace y dispositivos LoRaWAN. Para obtener más información, consulte[AWS IoTReferencia de la CLI inalámbrica](#).

AWS IoT Corefor LoRaWAN Regiones y puntos de enlace

AWS IoT Corepara LoraWAN proporciona soporte para extremos de API de plano de control y plano de datos específicos de suRegión de AWS. Los extremos de la API del plano de datos son específicos de suCuenta de AWSyRegión de AWS. Para obtener más información sobre elAWS IoT Corefor for LoRaWAN, consulte[AWS IoT Corefor LoRaWAN](#)en laAWSReferencia general de.

Para una comunicación más segura entre sus dispositivos yAWS IoT, puedes conectar tus dispositivos aAWS IoT Corefor LoRaWAN a través deAWS PrivateLinken su nube virtual privada (VPC), en lugar de conectarse a través de Internet pública. Para obtener más información, consulte [Conexión aAWS IoT Core for LoRaWANmediante un punto de enlace de interfaz de la VPC \(p. 1140\)](#).

AWS IoT Corepara LoraWAN tiene cuotas que se aplican a los datos del dispositivo que se transmiten entre los dispositivos y el TPS máximo para elAWS IoTOperaciones de la API inalámbrica. Para obtener más información, consulte[AWS IoT Corefor LoRaWAN](#)en laAWSReferencia general de.

AWS IoT Corefor LoRaWAN

Cuando te inscribes enAWS, puede comenzar conAWS IoT Corepara LoRawan sin cargo mediante el uso de[AWS Capa gratuita](#).

Para obtener más información acerca de la descripción general de los productos y los precios de, consulte[AWS IoT Corede IPAM](#).

¿Qué es ?AWS IoT Corefor LoRaWAN?

AWS IoT Corepara LoRaWAN reemplaza un servidor de red (LNS) privado LoRaWAN conectando sus dispositivos y puertas de enlace LoraWAN aAWS. Uso deAWS IoT, puede enrutar los mensajes recibidos desde dispositivos LoraWAN, donde se pueden formatear y enviar a otrosAWS IoTServicios de . Para proteger las comunicaciones de los dispositivos conAWS IoT,AWS IoT Corefor LoRaWAN utiliza certificados X.509.

AWS IoT Corefor LoraWAN administra las políticas de servicio y dispositivo queAWS IoT Corerequiere comunicarse con las puertas de enlace y dispositivos LoRaWAN.AWS IoT Corepara LoRaWan también gestiona los destinos que describen elAWS IoTReglas que envían datos de dispositivos a otros servicios.

conAWS IoT Corefor LoRaWAN, puede:

- Incorporar y conectar dispositivos y puertas de enlace LoraWAN aAWS IoTsin necesidad de configurar y administrar un LNS privado.
- Connect dispositivos LoRaWAN que cumplan las especificaciones 1.0.x o 1.1 LoRaWAN estandarizadas por LoRa Alianza. Estos dispositivos pueden funcionar en modo clase A, clase B o clase C.
- Utilizar puertas de enlace LoraWAN compatibles LoRa Basics Station versión 2.0.4 o posterior. Todas las puertas de enlace calificadas paraAWS IoT Corepara LoraWan ejecuta una versión compatible de LoRa Básicos de la estación.
- Supervise la intensidad de la señal, el ancho de banda y el factor de propagación medianteAWS IoT Corepara la velocidad de datos adaptativa de LoRaWan y optimiza la velocidad de datos si es necesario.
- Actualice el firmware de las puertas de enlace LoRaWAN mediante el servicio CUPS y el firmware de los dispositivos LoRaWAN mediante actualizaciones de firmware por aire (FUOTA).

Temas

- [¿Qué es LoRaWAN? \(p. 1118\)](#)
- [CómoAWS IoT Corefor LoRaWAN \(p. 1119\)](#)

¿Qué es LoRaWAN?

LaAlianza LoRadescribe a Lorawan como,«un protocolo de red de área amplia y de bajo consumo (LPWA) diseñado para conectar de forma inalámbrica «cosas» que funcionan con baterías a Internet en redes regionales, nacionales o globales, y se dirige a los requisitos clave de Internet de las cosas (IoT), como la comunicación bidireccional, end-to-end servicios de seguridad, movilidad y localización»..

LoRaWAN

El protocolo LoRaWAN es un protocolo de comunicación de redes de área amplia de bajo consumo (LPWAN) que funciona en LoRa. La especificación LoRaWAN está abierta para que cualquiera pueda configurar y operar un LoRa red.

LoRa es una tecnología de frecuencia de audio inalámbrica que funciona en un espectro de radiofrecuencia sin licencia. LoRa es un protocolo de capa física que utiliza modulación de espectro extendido y admite la comunicación de largo alcance a costa de un ancho de banda estrecho. Utiliza una forma de onda de banda estrecha con una frecuencia central para enviar datos, lo que hace que sean resistentes a las interferencias.

Características de la tecnología LoraWAN

- Comunicación de largo alcance de hasta 10 millas en línea de visión.
- Batería de larga duración de hasta 10 años. Para aumentar la duración de la batería, puede utilizar los dispositivos en modo clase A o clase B, lo que requiere una mayor latencia de enlace descendente.
- Bajo coste para dispositivos y mantenimiento.
- Espectro radioeléctrico sin licencia pero se aplican regulaciones específicas de cada región.
- Bajo consumo pero tiene un tamaño de carga útil limitado de 51 bytes a 241 bytes, dependiendo de la velocidad de datos. La velocidad de datos puede ser de 0,3 Kbit/s — 27 Kbit/s de velocidad de datos con un tamaño de carga útil máximo de 222.

Más información sobre LoRaWAN

Los siguientes enlaces contienen información útil sobre la tecnología LoraWAN y sobre LoRa Basics Station, que es el software que se ejecuta en las puertas de enlace LoraWAN para conectar dispositivos finales aAWS IoT Corefor LoRaWAN.

- [Fundamentos de las cosas en LoRawan](#)

The Things Fundamentals de LoRawan contiene un vídeo introductorio que cubre los fundamentos de LoraWAN y una serie de capítulos que te ayudarán a conocer LoRa y LoRaWAN.

- [¿Qué es LoRaWAN?](#)

LoRa Alliance proporciona una descripción técnica de LoRa y LoRawan, incluido un resumen de las especificaciones de LoRawan en diferentes regiones.

- [LoRa Básicos de la estación](#)

Semtech Corporation proporciona conceptos útiles sobre LoRa conceptos básicos para puertas de enlace y nodos finales. LoRa Basics Station, un software de código abierto que se ejecuta en su puerta de enlace LoraWAN, se mantiene y distribuye a través de laGitHub. También puede obtener información

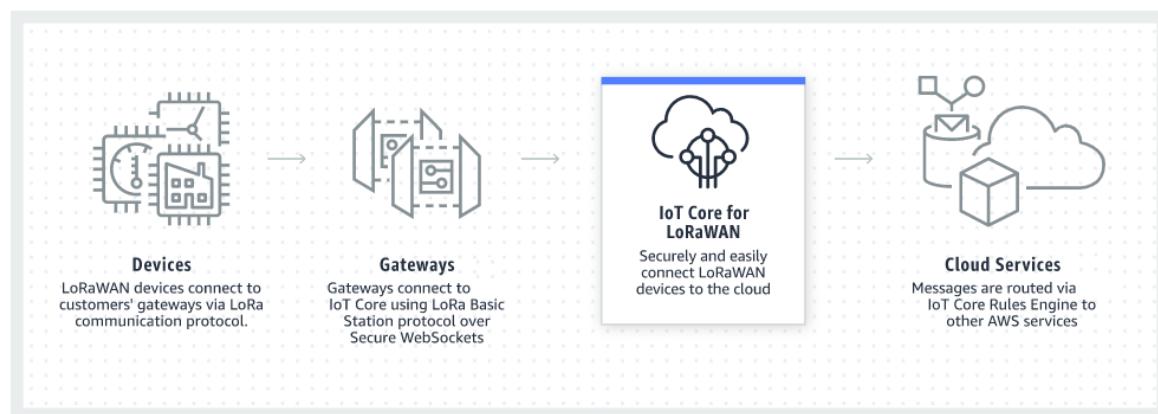
sobre los protocolos LNS y CUPS que describen cómo intercambiar datos de LoRaWAN y realizar actualizaciones de configuración.

CómoAWS IoT Corefor LoRaWAN

La arquitectura de red LoRaWAN se implementa iEn una estrella de la topología de estrellasy en la que las puertas de enlace transmiten información entre los dispositivos finales y el servidor de red LoRaWAN (LNS).

AWS IoT Corepara LoraWAN le ayuda a conectar y administrar dispositivos inalámbricos LoRaWAN (red de área amplia de largo alcance de bajo consumo) y reemplaza la necesidad de desarrollar y operar un LNS. Los dispositivos y puertas de enlace WAN de largo alcance (LoRaWAN) se pueden conectar aAWS IoT Coremediante el uso deAWS IoT Corefor LoRaWAN.

A continuación se muestra cómo interactúa un dispositivo LoraWAN conAWS IoT Corefor LoRaWAN. También se muestra cómoAWS IoT Corepara LoRaWan reemplaza un LNS y se comunica con otrosServicio de AWSs en elNube de AWS.



Los dispositivos LoRaWAN se comunican conAWS IoT Corea través de puertas de enlace LoRaWan.AWS IoT Corefor LoraWAN administra las políticas de servicio y dispositivo queAWS IoT Corerequiere administrar y comunicarse con las puertas de enlace y dispositivos LoRaWAN.AWS IoT Corepara LoRaWan también gestiona los destinos que describen elAWS IoTReglas que envían datos de dispositivos a otros servicios.

Comience a utilizarAWS IoT Corefor LoRaWAN

1. Seleccione los dispositivos inalámbricos y las puertas de enlace LoRaWAN que necesitará.

La[AWS Catálogo de dispositivos de socios](#)contiene gateways y kits de desarrollador que están calificados para su uso conAWS IoT Corefor LoRaWAN. Para obtener más información, consulte [Uso de puertas de enlace calificadas desde elAWS Catálogo de dispositivos asociados \(p. 1151\)](#).

2. Añada sus dispositivos inalámbricos y puertas de enlace LoRaWAN aAWS IoT Corefor LoRaWAN.

[Conexión de puertas de enlace y dispositivos aAWS IoT Core for LoRaWAN \(p. 1120\)](#)le proporciona información sobre cómo describir sus recursos y agregar dispositivos inalámbricos y puertas de enlace LoraWAN aAWS IoT Corefor LoRaWAN. También aprenderá a configurar el otroAWS IoT Corepara los recursos de LoraWAN que necesitará para administrar estos dispositivos y enviar sus datos aAWS Servicios de .

3. Completa tuAWS IoT Corefor LoRaWAN.

Empiece con[nuestra muestraAWS IoT Corefor LoRaWAN](#)y hazlo tuyo.

AWS IoT Corefor LoRaWAN

Los siguientes recursos le ayudarán a familiarizarse con la tecnología LoraWAN yAWS IoT Corefor LoRaWAN.

- [Introducción aAWS IoT Corefor LoRaWAN](#)

En el siguiente video se describe cómoAWS IoT Corepara LoraWAN funciona y le guía a través del proceso de adición de puertas de enlace LoRaWAN desde laAWS Management Console.

- [AWS IoT Corefor LoRaWAN](#)

El taller cubre los fundamentos de la tecnología LoRaWAN y su implementación conAWS IoT Corefor LoRaWAN. También puede utilizar el taller para recorrer los laboratorios que muestran cómo conectar la puerta de enlace y el dispositivo aAWS IoT Corepara LoRaWAN para crear una solución de IoT de muestra.

Conexión de puertas de enlace y dispositivos aAWS IoT Core for LoRaWAN

AWS IoT Core for LoRaWANle ayuda a conectarse y administrar la red inalámbricaLoRaWAN (red de área amplia de largo alcance de bajo consumo) y reemplaza la necesidad de desarrollar y operar un LNS. WAN de largo alcance (LoRaLos dispositivos y puertas de enlace (WAN) se pueden conectar aAWS IoT Coremediante el uso deAWS IoT Core for LoRaWAN.

Convenciones de nomenclatura para sus dispositivos, puertas de enlace, perfiles y destinos

Antes de empezar conAWS IoT Core for LoRaWANy crear los recursos, considere la convención de nomenclatura de sus dispositivos, puertas de enlace y destino.

AWS IoT Core for LoRaWANasigna ID exclusivos a los recursos que crea para dispositivos inalámbricos, puertas de enlace y perfiles; sin embargo, también puede dar a los recursos nombres más descriptivos para facilitar su identificación. Antes de agregar dispositivos, puertas de enlace, perfiles y destinos aAWS IoT Core for LoRaWAN, considere cómo les asignará un nombre para facilitar su administración.

También puede añadir etiquetas a los recursos que cree. Antes de añadir tuLoRaDispositivos WAN, considere cómo puede utilizar las etiquetas para identificar y administrar suAWS IoT Core for LoRaWANde AWS. Las etiquetas se pueden modificar después de agregarlas.

Para obtener más información sobre la denominación y el etiquetado, consulte[Describa susAWS IoT Corepara LoRaRecursos WAN \(p. 1121\)](#).

Asignación de datos de dispositivo a datos de servicio

Los datos deLoRaLos dispositivos inalámbricos WAN suelen codificarse para optimizar el ancho de banda. Estos mensajes codificados llegan aAWS IoT Core for LoRaWANen un formato que podría no ser utilizado fácilmente por otrosAWSservicios de .AWS IoT Core for LoRaWANutilizaAWS IoTReglas que pueden utilizarAWS Lambdafunciones para procesar y decodificar los mensajes del dispositivo en un formato que otrosAWSlos servicios pueden utilizar.

Para transformar los datos del dispositivo y enviarlos a otrosAWSservicios, necesita saber:

- Formato y contenido de los datos que envían los dispositivos inalámbricos.
- El servicio al que desea enviar los datos.
- Formato que requiere el servicio.

Con esa información, puede crear elAWS IoTregla que realiza la conversión y envía los datos convertidos alAWSservicios que lo utilizarán.

Uso de la consola para incorporar el dispositivo y la puerta de enlace aAWS IoT Core for LoRaWAN

Puede utilizar la interfaz de consola de o la API para agregar suLoRaPuerta de enlace y dispositivos WAN. Si utilizaAWS IoT Core for LoRaWANLe recomendamos que utilice la consola de por primera vez. La interfaz de la consola es más práctica cuando se administran unos cuantosAWS IoT Core for LoRaWANrecursos a la vez. Al administrar un gran número deAWS IoT Core for LoRaWANrecursos, considere la posibilidad de crear soluciones más automatizadas mediante elAWS IoT WirelessAPI.

Gran parte de los datos que introduce al configurarAWS IoT Core for LoRaWANlos proveedores de los dispositivos proporcionan los recursos y son específicos delLoRaEspecificaciones WAN que admiten. Los siguientes temas describen cómo puede describir suAWS IoT CoreparaLoRaRecursos WAN y utilice la consola o la API para agregar puertas de enlace y dispositivos.

Temas

- [Describa susAWS IoT CoreparaLoRaRecursos WAN \(p. 1121\)](#)
- [Incorporación de sus gateways aAWS IoT Core for LoRaWAN \(p. 1123\)](#)
- [Incorporación de sus dispositivos aAWS IoT Core for LoRaWAN \(p. 1129\)](#)

Describa susAWS IoT CoreparaLoRaRecursos WAN

Si utilizaAWS IoT Core for LoRaWANPor primera vez, puede añadir su primeraLoRaPuerta de enlace y dispositivo WAN mediante el[AWS IoT Core for LoRaWAN](#)Página de introducción deAWS IoTconsola de .

Antes de empezar a crear los recursos, considere la convención de nomenclatura de sus dispositivos, puertas de enlace y destino.AWS IoT Core for LoRaWANproporciona varias opciones para identificar los recursos que crea. Mientras queAWS IoT Core for LoRaWANlos recursos reciben un ID único cuando se crean, este ID no es descriptivo ni se puede cambiar después de crear el recurso. También puede asignar un nombre, añadir una descripción y adjuntar etiquetas y valores de etiqueta a la mayoríaAWS IoT Core for LoRaWANrecursos para que sea más conveniente seleccionar, identificar y administrar suAWS IoT Core for LoRaWANde AWS.

- [Nombres de recursos \(p. 1122\)](#)

Para puertas de enlace, dispositivos y perfiles, el nombre del recurso es un campo opcional que puede cambiar después de crear el recurso. El nombre aparece en las listas que se muestran en las páginas del centro de recursos.

En el caso de los destinos, proporciona un nombre único en suAWS cuenta de yRegión de AWS. No se puede cambiar el nombre del destino después de crear el recurso de destino.

Si bien un nombre puede tener hasta 256 caracteres, el espacio de visualización del centro de recursos es limitado. Asegúrese de que la parte distintiva del nombre aparezca en los primeros 20 a 30 caracteres, si es posible.

- [Etiquetas de recursos \(p. 1122\)](#)

Las etiquetas son pares de metadatos de clave-valor que se pueden asociar aAWSde AWS. Elige ambas claves de etiqueta y sus valores correspondientes.

Las puertas de enlace, los destinos y los perfiles pueden tener hasta 50 etiquetas adjuntas a ellos. Los dispositivos no admiten etiquetas.

Nombres de recursos

AWS IoT Core for LoRaWANcompatibilidad de recursos para name

Recurso	Compatibilidad con campos de nombres	
Destino	El nombre es un ID único de recurso y no se puede cambiar.	
Dispositivo	El nombre es un descriptor opcional del recurso y se puede cambiar.	
Puerta de enlace	El nombre es un descriptor opcional del recurso y se puede cambiar.	
Perfil	El nombre es un descriptor opcional del recurso y se puede cambiar.	

El campo nombre aparece en las listas de recursos del centro de recursos; sin embargo, el espacio es limitado y, por lo tanto, solo pueden verse los primeros 15-30 caracteres del nombre.

Al seleccionar nombres para los recursos, considere cómo desea que identifiquen los recursos y cómo se mostrarán en la consola.

Descripción

Los recursos de destino, dispositivo y puerta de enlace también admiten un campo de descripción, que puede aceptar hasta 2.048 caracteres. El campo de descripción aparece solo en la página de detalles del recurso individual. Aunque el campo de descripción puede contener mucha información, porque solo aparece en la página de detalles del recurso, no resulta conveniente escanear en el contexto de varios recursos.

Etiquetas de recursos

AWS IoT Core for LoRaWANsoporte de recursos paraAWSetiquetas

Recurso	AWSsoporte de etiquetas	
Destino	Hasta 50AWSse pueden agregar etiquetas al recurso.	
Dispositivo	Este recurso no admiteAWSetiquetas.	
Puerta de enlace	Hasta 50AWSse pueden agregar etiquetas al recurso.	

Recurso	AWSsoporte de etiquetas	
Perfil	Hasta 50AWSse pueden agregar etiquetas al recurso.	

Las etiquetas son palabras o frases que actúan como metadatos que puede utilizar para identificar y organizar suAWSde AWS. Puede pensar en la clave de etiqueta como una categoría de información y el valor de la etiqueta como un valor específico de esa categoría.

Por ejemplo, es posible que tenga un valor de etiqueta decolor, a continuación, dar a algunos recursos un valor deazulpara esa etiqueta y otros un valor dered. Con eso, podría usar la[Editor de etiquetas](#)en laAWSconsola para encontrar los recursos con uncolorvalor de etiqueta deazul.

Para obtener más información sobre estrategias de etiquetado y etiquetado, consulte[Editor de etiquetas](#).

Incorporación de sus gateways aAWS IoT Core for LoRaWAN

Si utilizaAWS IoT Core for LoRaWANPor primera vez, puede añadir su primeraLoRaPuerta de enlace y dispositivo WAN mediante la consola.

Antes de incorporar la puerta de enlace

Antes de incorporar la puerta de entrada aAWS IoT Core for LoRaWAN, le recomendamos que haga lo siguiente:

- Usar puertas de enlace que estén calificadas para su uso conAWS IoT Core for LoRaWAN. Estas puertas de enlace se conectan aAWS IoT Coresin ajustes de configuración adicionales y tienen una versión compatible del [LoRaBasics Station](#)software que se ejecuta en ellos. Para obtener más información, consulte [Administración de gateways conAWS IoT Corefor LoRaWAN \(p. 1150\)](#).
- Considere la convención de nomenclatura de los recursos que crea para que pueda administrarlos con mayor facilidad. Para obtener más información, consulte [Describa susAWS IoT CoreparaLoRaRecursos WAN \(p. 1121\)](#).
- Tenga listos los parámetros de configuración exclusivos de cada puerta de enlace para ingresar por adelantado, lo que hace que la introducción de los datos en la consola sea más fluida. Parámetros de configuración de puerta de enlace inalámbrica queAWS IoT requiere comunicarse con la puerta de enlace y administrar la puerta de enlace, incluir la EUI de la puerta de enlace y suLoRabanda de frecuencia.

Para incorporar sus puertas de enlace aAWS IoT Core for LoRaWAN:

- [Considere la selección de bandas de frecuencia y agregue la función IAM necesaria \(p. 1123\)](#)
- [Agregar una puerta de enlace aAWS IoT Core for LoRaWAN \(p. 1125\)](#)
- [Connect elLoRaPuerta de enlace WAN y verificar el estado de su conexión \(p. 1128\)](#)

Considere la selección de bandas de frecuencia y agregue la función IAM necesaria

Antes de agregar la puerta de enlace aAWS IoT Core for LoRaWAN, le recomendamos que considere la banda de frecuencia en la que operará la puerta de enlace y agregue la función de IAM necesaria para conectar la puerta de enlace aAWS IoT Core for LoRaWAN.

Note

Si va a agregar la gateway mediante la consola de, haga clic enCreación de un rol en la consola para crear el rol de IAM necesario para que pueda omitir estos pasos. Debe realizar estos pasos solo si está utilizando la CLI para crear la puerta de enlace.

Considerar la selección deLoRabandas de frecuencia para las puertas de enlace y la conexión de dispositivos

AWS IoT Core for LoRaWANadmite bandas de frecuencia EU863-870, US902-928, AU915 y AS923-1, que puede utilizar para conectar las puertas de enlace y dispositivos que están físicamente presentes en países que admiten los rangos de frecuencia y las características de estas bandas. Las bandas EU863-870 y US902-928 se utilizan comúnmente en Europa y Norteamérica, respectivamente. La banda AS923-1 se usa comúnmente en Australia, Nueva Zelanda, Japón y Singapur, entre otros países. El AU915 se utiliza en Australia y Argentina, entre otros países. Para obtener más información acerca de qué banda de frecuencias debe usar en su región o país, consulte [LoRaParámetros regionales de WAN®](#).

LoRaAlliance publicaLoRaEspecificaciones WAN y documentos de parámetros regionales que se encuentran disponibles para descargarlas enLoRaSítioweb de Alliance. LaLoRaLos parámetros regionales de Alliance ayudan a las empresas a decidir qué banda de frecuencia utilizar en su región o país.AWS IoT Core for LoRaWANla implementación de la banda de frecuencia sigue la recomendación del documento de especificación de parámetros regionales. Estos parámetros regionales se agrupan en un conjunto de parámetros de radio, junto con una asignación de frecuencia adaptada a la banda industrial, científica y médica (ISM). Le recomendamos que trabaje con los equipos de cumplimiento para asegurarse de cumplir con los requisitos normativos aplicables.

Agregar un rol de IAM para permitir que el Servidor de configuración y actualización (CUPS) administre las credenciales de puerta de enlace

En este procedimiento se describe cómo agregar un rol de IAM que permitirá al Servidor de configuración y actualización (CUPS) administrar las credenciales de puerta de enlace. Asegúrese de realizar este procedimiento antes de unLoRaLa puerta de enlace WAN intenta conectarse conAWS IoT Core for LoRaWANSin embargo, solo es necesario hacerlo una vez.

Agregue el rol de IAM para permitir que el Servidor de configuración y actualización (CUPS) administre las credenciales de puerta de enlace

1. Abra el iconoCentro de roles de la consola de IAMy eligeCreación de un rol.
2. Si cree que puede que ya haya agregado elloTWirelessGatewayCertManagerRolrole, en la barra de búsqueda, escriba**IoTWirelessGatewayCertManagerRole**.

Si ve unloTWirelessGatewayCertManagerRolen los resultados de búsqueda, tiene el rol de IAM necesario. Puede dejar el procedimiento ahora.

Si los resultados de búsqueda están vacíos, no tiene el rol de IAM necesario. Continúe con el procedimiento para agregarlo.

3. EnSeleccione el tipo de entidad de confianza, eligeOtroCuenta de AWS.
4. EnID de cuenta, introduzca suCuenta de AWSID y, a continuación, elijaSiguiente: Permisos.
5. En el cuadro de búsqueda, escriba **AWSIoTWirelessGatewayCertManager**.
6. En la lista de resultados de búsqueda, seleccione la política denominadaAWSIoTWirelessGatewayCertManager.
7. Seleccione Next (Siguiente): Etiquetas luego seleccioneSiguiente: Consulte.
8. EnNombre del rol, introduzca**IoTWirelessGatewayCertManagerRole**y luego seleccioneCreación de un rol.

9. Para editar el nuevo rol, en el mensaje de confirmación, elija **IoTWirelessGatewayCertManagerRole**.
10. En **Resumen**, elige **Relaciones de confianza**, y, a continuación, elija **Modificar relación de confianza**.
11. En **Policy Document**, cambie la **Principal** a similar a este ejemplo.

```
"Principal": {  
    "Service": "iotwireless.amazonaws.com"  
},
```

Después de cambiar el **Principal**, el documento de política completo debe tener el aspecto del siguiente ejemplo.

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

12. Para guardar los cambios y salir, elija **Actualización de política de confianza**.

Ya has creado **IoTWirelessGatewayCertManagerRole**. No tendrás que volver a hacerlo.

Si ha realizado este procedimiento mientras agregaba una puerta de enlace, puede cerrar esta ventana y la consola de IAM y volver a la AWS IoT consola para terminar de agregar la puerta de enlace.

Agregar una puerta de enlace a AWS IoT Core for LoRaWAN

Puedes añadir tu puerta de enlace a AWS IoT Core for LoRaWAN utilizando la consola o la CLI.

Antes de agregar su gateway, le recomendamos que tenga en cuenta los factores mencionados en [el Antes de incorporar la puerta de enlace](#) sección de [Incorporación de sus gateways a AWS IoT Core for LoRaWAN \(p. 1123\)](#).

Si va a agregar la puerta de enlace por primera vez, se recomienda utilizar la consola. Si desea agregar la puerta de enlace mediante la CLI en su lugar, debe haber creado el rol de IAM necesario para que la puerta de enlace pueda conectarse con AWS IoT Core for LoRaWAN. Para obtener información sobre cómo crear el rol, consulte [Aregar un rol de IAM para permitir que el Servidor de configuración y actualización \(CUPS\) administre las credenciales de puerta de enlace \(p. 1124\)](#).

Añada una gateway con la consola

Vaya a la [AWS IoT Core for LoRaWAN](#) Intro Página de la AWS IoT consola de y elija **Introducción** luego seleccione **Añadir gateway**. Si ya has agregado una puerta de enlace, elige **Ver puerta de enlace** para ver la puerta de enlace que ha agregado. Si desea agregar más puertas de enlace, elija **Añadir gateway**.

1. Proporcionar detalles de pasarela e información de banda de frecuencia

Usa **Detalles de pasarela** para proporcionar información sobre los datos de configuración del dispositivo, como la EUI de la puerta de enlace y la configuración de banda de frecuencia.

- EUI de Gateway

EUI (identificador único extendido) del dispositivo de puerta de enlace individual. La EUI es un código alfanumérico de 16 dígitos, como c0ee40fffff29df10, que identifica de forma exclusiva una puerta de enlace en suLoRaRed WAN. Esta información es específica del modelo de puerta de enlace y puede encontrarla en su dispositivo de puerta de enlace o en su manual de usuario.

Note

La EUI de la puerta de enlace es diferente de la dirección MAC Wi-Fi que puede ver impresa en su dispositivo de puerta de enlace. La EUI sigue un estándar EUI-64 que identifica de forma exclusiva su puerta de enlace y, por lo tanto, no se puede volver a utilizar en otrasCuenta de AWSs y regiones.

- Banda de frecuencia (región)

La banda de frecuencia de la puerta de enlace. Puede elegir entre US915, EU868, AU915, o bien AS923-1, según el soporte de la puerta de enlace y del país o región desde el que se conecte físicamente la puerta de enlace. Para obtener más información sobre las bandas, consulte [Considerar la selección deLoRabandas de frecuencia para las puertas de enlace y la conexión de dispositivos \(p. 1124\)](#).

2. Especificar los datos de configuración de la puerta de enlace inalámbrica (opcional)

Estos campos son opcionales y puede utilizarlos para proporcionar información adicional sobre la puerta de enlace y su configuración.

- Nombre, descripción y etiquetas de la puerta de enlace

La información de estos camposopcionales proviene de cómo organiza y describe los elementos de su sistema inalámbrico. Puede asignar unNombre a la puerta de enlace, utilice elDescripción para proporcionar información sobre la puerta de enlace y utilizarEtiquetas para añadir pares de metadatos clave-valor sobre la puerta de enlace. Para obtener más información sobre cómo asignar nombres y describir los recursos de, consulte [Describa susAWS IoT CoreparaLoRaRecursos WAN \(p. 1121\)](#).

- LoRaConfiguración WAN mediante subbandas y filtros

Si lo desea, también puede especificarLoRaDatos de configuración de WAN, como las subbandas que desea utilizar y los filtros que pueden controlar el flujo de tráfico. Para este tutorial, puedes omitir estos campos. Para obtener más información, consulte [Configurar las subbandas y las capacidades de filtrado de la puerta de enlace \(p. 1151\)](#).

3. Asociar unAWS IoTcosa con la puerta de enlace

Especificar si se va a crear unAWS IoTy asociarlo con la puerta de enlace. Cosas deAWS IoTpermite facilitar la búsqueda y la administración de los dispositivos de. La asociación de una cosa a la puerta de enlace permite que la puerta de enlace acceda a otrosAWS IoT Corecaracterísticas.

4. Crear y descargar el certificado de puerta de enlace

Para autenticar la puerta de enlace para que pueda comunicarse de forma segura conAWS IoT, suLoRaLa puerta de enlace WAN debe presentar una clave privada y un certificado paraAWS IoT Core for LoRaWAN. Creación de unCertificado de pasarelade modo queAWS IoTpuede verificar la identidad de la puerta de enlace mediante el estándar X.509.

Haga clic en el botón .Crear un certificadoy descargue los archivos de certificado. Los usarás más tarde para configurar la puerta de enlace.

5. Copie los endpoints CUPS y LNS y descargue certificados

SusLoRaLa puerta de enlace WAN debe conectarse a un extremo CUPS o LNS al establecer una conexión aAWS IoT Core for LoRaWAN. Recomendamos utilizar el punto final CUPS ya que también proporciona administración de la configuración. Para verificar la autenticidad deAWS IoT Core for LoRaWANendpoints, la puerta de enlace utilizará un certificado de confianza para cada uno de los endpoints CUPS y LNS,

Haga clic en el botón .Copiapara copiar los endpoints CUPS y LNS. Necesitará esta información más tarde para configurar la gateway. Luego haga clic en la teclaDescargar certificados de confianza del servidorpara descargar los certificados de confianza de los endpoints CUPS y LNS.

6. Crear el rol de IAM para los permisos de puerta de enlace

Debe agregar un rol de IAM que permita al Servidor de configuración y actualización (CUPS) administrar las credenciales de puerta de enlace. Debe hacerlo antes de unLoRaLa puerta de enlace WAN intenta conectarse conAWS IoT Core for LoRaWAN; sin embargo, solo tienes que hacerlo una vez.

Para crear elloTWirelessGatewayCertManagerFunción de IAM para su cuenta, haga clic en laCreación de un rolBotón. Si el rol ya existe, selecciónelo en la lista desplegable.

ClicEnviarpara completar la creación de la puerta de enlace.

Agregar una puerta de enlace mediante la API

Si va a agregar una puerta de enlace por primera vez mediante la API o la CLI, debe agregar elloTWirelessGatewayCertManagerFunción de IAM para que la puerta de enlace pueda conectarse conAWS IoT Core for LoRaWAN. Para obtener información acerca de cómo crear el rol, consulte la siguiente secciónAregar un rol de IAM para permitir que el Servidor de configuración y actualización (CUPS) administre las credenciales de puerta de enlace (p. 1124).

En las listas siguientes se describen las acciones de la API que realizan las tareas asociadas con la adición, actualización o eliminación de unLoRaGateway WAN.

AWS IoT WirelessAcciones de API paraAWS IoT Core for LoRaWANGateways de

- [CreateWirelessPortal](#)
- [GetWirelessPortal](#)
- [ListWirelessGateways de](#)
- [UpdateWirelessPortal](#)
- [DeleteWirelessPortal](#)

Para ver la lista completa de acciones y tipos de datos disponibles para crear y administrarAWS IoT Core for LoRaWANrecursos, consulte la[AWS IoT WirelessReferencia de la API de](#).

Cómo utilizar laAWS CLIpasarregar una puerta de enlace

Puede utilizar elAWS CLIpasarrear una puerta de enlace inalámbrica mediante el[create-wireless-gateway](#) comando. En el ejemplo siguiente se crea un dispositivo inalámbrico.LoRaPasarela de dispositivos WAN. También puede proporcionar un que contendrá detalles adicionales, como el certificado de puerta de enlace y las credenciales de aprovisionamiento.

Note

También puede realizar este procedimiento con la API utilizando los métodos de la API de AWS que corresponden a los comandos CLI que se muestran aquí.

```
aws iotwireless create-wireless-gateway \
--lorawan GatewayEui="a1b2c3d4567890ab",RfRegion="US915" \
--name "myFirstLoRaWANGateway" \
--description "Using my first LoRaWAN gateway"
--cli-input-json input.json
```

Para obtener información sobre los CLI que puede usar, consulte[AWS CLIReferencia de de](#)

Connect elLoRaPuerta de enlace WAN y verificar el estado de su conexión

Antes de poder comprobar el estado de la conexión de la puerta de enlace, debe haber agregado ya la puerta de enlace y conectarla aAWS IoT Core for LoRaWAN. Para obtener información sobre cómo agregar la gateway, consulte [Agregar una puerta de enlace aAWS IoT Core for LoRaWAN \(p. 1125\)](#).

Connect su gateway aAWS IoT Core for LoRaWAN

Después de agregar la puerta de enlace, conéctese a la interfaz de configuración de la puerta de enlace para introducir la información de configuración y los certificados de confianza.

Después de añadir la información de la puerta de enlace aAWS IoT Core for LoRaWAN, añadir un pocoAWS IoT Core for LoRaWANinformación al dispositivo gateway. La documentación proporcionada por el proveedor de la puerta de enlace debe describir el proceso de carga de los archivos de certificado en la puerta de enlace y configurar el dispositivo de puerta de enlace para comunicarse conAWS IoT Core for LoRaWAN.

Pasarelas calificadas para su uso conAWS IoT Core for LoRaWAN

Para obtener instrucciones sobre cómo configurar suLoRaPuerta de enlace WAN, consulte la[configurar gateway device](#)Sección sobre de laAWS IoT Core for LoRaWANtaller. Aquí encontrará información sobre las instrucciones para conectar puertas de enlace que son aptas para su uso conAWS IoT Core for LoRaWAN.

Pasarelas compatibles con el protocolo CUPS

Las siguientes instrucciones muestran cómo puede conectar las puertas de enlace que admiten el protocolo CUPS.

1. Cargue los siguientes archivos que obtuvo al agregar la puerta de enlace.
 - Certificado de dispositivo de gateway y archivos de clave privada.
 - Archivo de certificado de confianza para endpoint CUPS,`cups.trust`.
2. Especifique la URL del endpoint CUPS que ha obtenido anteriormente. El punto final tendrá el formato`prefix.cups.lorawan.region.amazonaws.com:443`.

Para obtener información detallada sobre cómo obtener esta información, consulte [Agregar una puerta de enlace aAWS IoT Core for LoRaWAN \(p. 1125\)](#).

Gateways compatibles con el protocolo LNS

Las siguientes instrucciones muestran cómo puede conectar las puertas de enlace que admiten el protocolo LNS.

1. Cargue los siguientes archivos que obtuvo al agregar la puerta de enlace.
 - Certificado de dispositivo de gateway y archivos de clave privada.
 - Archivo de certificado de confianza para endpoint LNS,`lns.trust`.
2. Especifique la URL del endpoint LNS que ha obtenido anteriormente. El punto final tendrá el formato`prefix.lns.lorawan.region.amazonaws.com:443`.

Para obtener información detallada sobre cómo obtener esta información, consulte [Agregar una puerta de enlace aAWS IoT Core for LoRaWAN \(p. 1125\)](#).

Después de eso, has conectado tu puerta de enlace aAWS IoT Core for LoRaWAN, puedes comprobar el estado de la conexión y obtener información acerca de cuándo se recibió el último vínculo ascendente mediante la consola o la API.

Comprobar el estado de la conexión de la puerta de enlace

Para comprobar el estado de la conexión mediante la consola, navegue hasta la[Gateways de Página de la AWS IoT consola](#) y elige la puerta de enlace que has agregado. En el navegadorLoRaDetalles específicos de WAN de la página de detalles de la puerta de enlace, verá el estado de la conexión y la fecha y hora en que se recibió el último enlace ascendente.

Comprobar el estado de la conexión de gateway mediante la API

Para comprobar el estado de la conexión mediante la API, utilice `laGetWirelessGatewayStatisticsAPI`. Esta API no tiene cuerpo de solicitud y solo contiene un cuerpo de respuesta que muestra si la puerta de enlace está conectada y cuándo se recibió el último vínculo ascendente.

```
HTTP/1.1 200
Content-type: application/json

{
    "ConnectionStatus": "Connected",
    "LastUplinkReceivedAt": "2021-03-24T23:13:08.476015749Z",
    "WirelessGatewayId": "30cbdcf3-86de-4291-bfab-5bfa2b12bad5"
}
```

Incorporación de sus dispositivos aAWS IoT Core for LoRaWAN

Después de haber incorporado su puerta de enlace aAWS IoT Core for LoRaWANy verificó su estado de conexión, puede incorporar sus dispositivos inalámbricos. Para obtener información sobre cómo incorporar las puertas de enlace, consulte[Incorporación de sus gateways aAWS IoT Core for LoRaWAN \(p. 1123\)](#).

Los dispositivos LoRaWAN utilizan unProtocolo LoRaWAN para intercambiar datos con aplicaciones alojadas en la nube.AWS IoT Core for LoRaWANadmite dispositivos que cumplen los requisitos 1.0.x o 1.1LoRaEspecificaciones Wan estandarizadas porLoRaAlianza.

El dispositivo LoRaWAN suele contener uno o varios sensores y actores. Los dispositivos envían datos de telemetría de enlace ascendente a través deLoRaPasarelas WAN aAWS IoT Core for LoRaWAN. Las aplicaciones alojadas en la nube pueden controlar los sensores enviando comandos de enlace descendente aLoRaDispositivos WAN medianteLoRaPuertas de enlace WAN.

Antes de incorporar el dispositivo inalámbrico

Antes de incorporar el dispositivo inalámbrico aAWS IoT Core for LoRaWAN, debe tener lista la siguiente información con antelación:

- LoRaEspecificación Wan y configuración de dispositivos inalámbricos

Tener los parámetros de configuración exclusivos de cada dispositivo listos para entrar por adelantado hace que introducir los datos en la consola sea más fluida. Los parámetros específicos que debe introducir dependen deLoRaEspecificación Wan que utiliza el dispositivo. Para obtener una lista completa de sus especificaciones y parámetros de configuración, consulte la documentación de cada dispositivo.

- Nombre y descripción del dispositivo (opcional)

La información de estos campos opcionales proviene de cómo organiza y describe los elementos de su sistema inalámbrico. Para obtener más información acerca de cómo asignar nombres y describir sus recursos, consulte[Describa sus AWS IoT Core para LoRaRecursos Wan \(p. 1121\)](#).

- Perfiles de dispositivo y servicio

Tener algunos parámetros de configuración de dispositivos inalámbricos listos que son compartidos por muchos dispositivos y se pueden almacenar en AWS IoT Core for LoRaWAN como perfiles de dispositivos y servicios. Los parámetros de configuración se encuentran en la documentación del dispositivo o en el dispositivo. Querrás identificar un perfil de dispositivo que coincida con los parámetros de configuración del dispositivo o crear uno si es necesario, antes de añadir el dispositivo. Para obtener más información, consulte [Aregar perfiles aAWS IoT Core for LoRaWAN \(p. 1132\)](#).

- AWS IoT Core for LoRaWAN destino

Cada dispositivo debe asignarse a un destino que procesará sus mensajes para enviarlos a AWS IoT. Los otros servicios. La AWS IoT las reglas que procesan y envían los mensajes del dispositivo son específicas del formato de mensaje del dispositivo. Para procesar los mensajes desde el dispositivo y enviarlos al servicio correcto, identifique el destino que creará para usar con los mensajes del dispositivo y asígnelo al dispositivo.

Para incorporar su dispositivo inalámbrico en AWS IoT Core for LoRaWAN

- [Añade tu dispositivo inalámbrico aAWS IoT Core for LoRaWAN \(p. 1130\)](#)
- [Aregar perfiles aAWS IoT Core for LoRaWAN \(p. 1132\)](#)
- [Añadir destinos aAWS IoT Core para LoRaWAN \(p. 1134\)](#)
- [Crear reglas para procesar mensajes de dispositivos WAN \(p. 1137\)](#)
- [Connect el dispositivo WAN y verifique el estado de su conexión \(p. 1139\)](#)

Añade tu dispositivo inalámbrico aAWS IoT Core for LoRaWAN

Si va a agregar su dispositivo inalámbrico por primera vez, se recomienda utilizar la consola. Vaya a la [AWS IoT Core for LoRaWAN Intro](#) Página de la AWS IoT consola de, seleccione [Introducción](#) y luego seleccione [Añadir dispositivo](#). Si ya ha añadido un dispositivo, elige [Dispositivo](#) de visualización para ver la puerta de enlace que ha agregado. Si desea añadir más dispositivos, elija [Añadir dispositivo](#).

De otra manera, también puede añadir dispositivos inalámbricos desde la [Dispositivos](#) Página de la AWS IoT consola de .

Añada la especificación de su dispositivo inalámbrico aAWS IoT Core for LoRaWAN Uso de la consola

Seleccione una especificación de dispositivo inalámbrico según el método de activación y el LoRaWAN Versión. Una vez seleccionados, los datos se cifran con una clave que AWS posee y administra por ti.

Modos de activación OTAA y ABP

Antes de que el dispositivo WAN pueda enviar datos de enlace ascendente, debe completar un proceso llamado [activación](#) o [procedimiento de unión](#). Para activar el dispositivo, puede utilizar OTAA (activación por aire) o ABP (activación por personalización).

ABP no requiere un procedimiento de unión y utiliza claves estáticas. Cuando usas OTAA, tu dispositivo WAN envía una solicitud de unión y el servidor de red puede permitir la solicitud. Le recomendamos que utilice OTAA para activar el dispositivo porque se generan nuevas claves de sesión para cada activación, lo que lo hace más seguro.

LoRaWAN Versión WAN

Cuando usas OTAA, tu dispositivo WAN y las aplicaciones alojadas en la nube comparten las claves raíz. Estas claves raíz dependen de si usas la versión v1.0.x o la v1.1. v1.0.x solo tiene una clave raíz, AppKey(Clave de aplicación) mientras que la v1.1 tiene dos claves raíz, AppKey(clave de aplicación)

yNwkKey(Clave de red). Las claves de sesión se derivan en función de las claves raíz de cada activación. AmbosNwkKeyyAppKeyson valores hexadecimales de 32 dígitos proporcionados por su proveedor de servicios inalámbricos.

IU de dispositivos inalámbricos

Después de seleccionar elEspecificación de dispositivo inalámbrico, verá los parámetros EUI (identificador único extendido) del dispositivo inalámbrico que se muestran en la consola. Puede encontrar esta información en la documentación del dispositivo o del proveedor de servicios inalámbricos.

- DevEui: valor hexadémico de 16 dígitos exclusivo de su dispositivo y que se encuentra en la etiqueta del dispositivo o en su documentación.
- ApEui: valor hexadémico de 16 dígitos exclusivo del servidor de unión y que se encuentra en la documentación del dispositivo. EnLoRaWAN versión 1.1, laApEuis llamadoJoinEui.

Para obtener más información sobre los identificadores exclusivos, las claves de sesión y las claves raíz, consulte la [LoRaAlianza](#).

Añada la especificación de su dispositivo inalámbrico aAWS IoT Core for LoRaWANmediante la API

Si vas a agregar un dispositivo inalámbrico mediante la API, debes crear el perfil de dispositivo y el perfil de servicio antes de crear el dispositivo inalámbrico. Utilizarás el perfil de dispositivo y el identificador del perfil de servicio al crear el dispositivo inalámbrico. Para obtener información acerca de cómo crear estos perfiles con la API, consulte[Agregar un perfil de dispositivo mediante la API \(p. 1132\)](#).

En las listas siguientes se describen las acciones de la API que realizan las tareas asociadas con la adición, actualización o eliminación de un perfil de servicio.

AWS IoT WirelessAcciones de API para perfiles de servicio

- [CreateWirelessDispositivo](#)
- [GetWirelessDispositivo](#)
- [ListWirelessDispositivos](#)
- [UpdateWirelessDispositivo](#)
- [DeleteWirelessDispositivo](#)

Para ver la lista completa de acciones y tipos de datos disponibles para crear y administrarAWS IoT Core for LoRaWANrecursos, consulte la[AWS IoT WirelessReferencia de la API de](#).

Cómo utilizar laAWS CLIp para crear un dispositivo inalámbrico

Puede utilizar elAWS CLIpara crear un dispositivo inalámbrico mediante elcreate-wireless-devicecomando. En el ejemplo siguiente se crea un dispositivo inalámbrico mediante un archivo.json de entrada para introducir los parámetros.

Note

También puede realizar este procedimiento con la API utilizando los métodos de la API de AWS que corresponden a los comandos CLI que se muestran aquí.

Contenido de 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
}
```

Puede proporcionar este archivo como entrada a la `create-wireless-device` comando.

```
aws iotwireless create-wireless-device \
--cli-input-json file://input.json
```

Para obtener información sobre los CLI que puede usar, consulte [AWS CLI Referencia de de](#)

Agregar perfiles aAWS IoT Core for LoRaWAN

Los perfiles de dispositivos y servicios se pueden definir para describir configuraciones de dispositivos comunes. Estos perfiles describen los parámetros de configuración que comparten los dispositivos para facilitar la adición de esos dispositivos. AWS IoT Core for LoRaWAN admite perfiles de dispositivos y perfiles de servicio.

El fabricante del dispositivo proporciona los parámetros de configuración y los valores que se deben introducir en estos perfiles.

Agregar perfiles de dispositivo

Los perfiles de dispositivo definen las capacidades del dispositivo y los parámetros de arranque que utiliza el servidor de red para configurar el LoRaServicio de acceso de radio WAN. Incluye selección de parámetros tales como LoRaBanda de frecuencia, LoRaRversión de parámetros regionales y versión MAC del dispositivo. Para obtener información sobre las diferentes bandas de frecuencia, consulte [Considerar la selección de LoRaBandas de frecuencia para las puertas de enlace y la conexión de dispositivos \(p. 1124\)](#).

Agregar un perfil de dispositivo mediante la consola

Si va a agregar un dispositivo inalámbrico mediante la consola, tal y como se describe en [Añada la especificación de su dispositivo inalámbrico aAWS IoT Core for LoRaWAN](#) [Uso de la consola \(p. 1130\)](#), una vez que hayas añadido la especificación del dispositivo inalámbrico, puedes añadir tu perfil de dispositivo. De otra manera, también puedes añadir dispositivos inalámbricos desde la [Página de la AWS IoT consola](#) en la [LoRaWAN Pestaña](#).

Puede elegir entre los perfiles de dispositivo predeterminados o crear un nuevo perfil de dispositivo. Le recomendamos que utilice los perfiles de dispositivo predeterminados. Si la aplicación requiere que cree un perfil de dispositivo, proporcione un Nombre de perfil de dispositivo, seleccione la Banda de frecuencia (RfRegion) que utilizas para el dispositivo y la puerta de enlace, y mantiene los demás ajustes a los valores predeterminados, a menos que se especifique lo contrario en la documentación del dispositivo.

Agregar un perfil de dispositivo mediante la API

Si vas a agregar un dispositivo inalámbrico mediante la API, debes crear tu perfil de dispositivo antes de crear el dispositivo inalámbrico.

En las listas siguientes se describen las acciones de la API que realizan las tareas asociadas con la adición, actualización o eliminación de un perfil de servicio.

AWS IoT WirelessAcciones de API para perfiles de servicio

- [CreateDevicePerfil de](#)
- [GetDevicePerfil de](#)
- [ListDeviceProfiles](#)
- [UpdateDevicePerfil de](#)
- [DeleteDevicePerfil de](#)

Para ver la lista completa de acciones y tipos de datos disponibles para crear y administrar AWS IoT Core for LoRaWAN recursos, consulte la [AWS IoT Wireless Referencia de la API de](#).

Cómo utilizar la AWS CLI para crear un perfil de dispositivo

Puede utilizar la AWS CLI para crear un perfil de dispositivo mediante la [create-device-profile](#) comando. En el ejemplo siguiente se crea un perfil de dispositivo.

```
aws iotwireless create-device-profile
```

Al ejecutar este comando se crea automáticamente un perfil de dispositivo con un ID que puede utilizar al crear el dispositivo inalámbrico. Ahora puede crear el perfil de servicio mediante la siguiente API y, a continuación, crear el dispositivo inalámbrico mediante los perfiles de dispositivo y servicio.

```
{  
    "Arn": "arn:aws:iotwireless:us-east-1:123456789012:DeviceProfile/12345678-a1b2-3c45-67d8-e90fa1b2c34d",  
    "Id": "12345678-a1b2-3c45-67d8-e90fa1b2c34d"  
}
```

Para obtener información sobre los CLI que puede usar, consulte [AWS CLI Referencia de de](#)

Añada perfiles de servicio

Los perfiles de servicio describen los parámetros de comunicación que necesita el dispositivo para comunicarse con el servidor de aplicaciones.

Agregar un perfil de servicio mediante la consola

Si va a agregar un dispositivo inalámbrico mediante la consola, tal y como se describe en [Añada la especificación de su dispositivo inalámbrico a AWS IoT Core for LoRaWAN Uso de la consola \(p. 1130\)](#), después de agregar el perfil del dispositivo, puede agregar su perfil de servicio. De otra manera, también puede añadir dispositivos inalámbricos desde la [Perfiles Página de la AWS IoT consola en la LoRaWAN Pestaña](#).

Le recomendamos que deje la configuración. Añadir GWMetaData habilitado para que reciba metadatos de puerta de enlace adicionales para cada carga útil, como RSSI y SNR para la transmisión de datos.

Agregar un perfil de servicio mediante la API

Si va a agregar un dispositivo inalámbrico mediante la API, primero debe crear su perfil de servicio antes de crear el dispositivo inalámbrico.

En las listas siguientes se describen las acciones de la API que realizan las tareas asociadas con la adición, actualización o eliminación de un perfil de servicio.

AWS IoT WirelessAcciones de API para perfiles de servicio

- [CreateServicePerfil de](#)
- [GetServicePerfil de](#)
- [ListServicePerfiles](#)
- [UpdateServicePerfil de](#)
- [DeleteServicePerfil de](#)

Para ver la lista completa de acciones y tipos de datos disponibles para crear y administrar AWS IoT Core for LoRaWAN recursos, consulte la [AWS IoT WirelessReferencia de la API de](#).

Cómo utilizar la AWS CLI para crear un perfil de servicio

Puede utilizar la AWS CLI para crear un servicio utilizando la [create-service-profile](#) comando. En el ejemplo siguiente se crea un perfil de servicio.

```
aws iotwireless create-service-profile
```

Al ejecutar este comando se crea automáticamente un perfil de servicio con un ID que puede utilizar al crear el dispositivo inalámbrico. Ahora puede crear el dispositivo inalámbrico utilizando el dispositivo y los perfiles de servicio.

```
{  
    "Arn": "arn:aws:iotwireless:us-east-1:123456789012:ServiceProfile/12345678-a1b2-3c45-67d8-e90fa1b2c34d",  
    "Id": "12345678-a1b2-3c45-67d8-e90fa1b2c34d"  
}
```

Añadir destinos a AWS IoT Core para LoRaWAN

AWS IoT Core para LoRaWAN Los destinos WAN describen el AWS IoT que procesa los datos de un dispositivo para utilizarlos por AWS Services de .

Porque la mayoría de los dispositivos WAN no envían datos a AWS IoT Core para LoRaWAN en un formato que puede utilizar AWS Services, una AWS IoT regla debe procesarla primero. La AWS IoT contiene la instrucción SQL que interpreta los datos del dispositivo y las acciones de reglas de tema que envían el resultado de la sentencia SQL a los servicios que la utilizarán.

Si va a agregar su destino por primera vez, se recomienda utilizar la consola.

Añada un destino con la consola

Si va a agregar un dispositivo inalámbrico mediante la consola, tal y como se describe en [Añade la especificación de su dispositivo inalámbrico a AWS IoT Core for LoRaWAN](#) Uso de la consola (p. 1130), después de haber agregado la especificación y los perfiles del dispositivo inalámbrico a AWS IoT Core for LoRaWAN como se ha descrito anteriormente, puede seguir adelante y añadir un destino.

También puede agregar un AWS IoT Core for LoRaWAN destino desde el Destinos Página de la AWS IoT consola de .

Para procesar los datos de un dispositivo, especifique los siguientes campos al crear un AWS IoT Core para LoRaWAN destino WAN y, a continuación, elija Agregar destino.

- Detalles del destino

Escriba un Nombre de destino y una descripción opcional para su destino.

- Nombre de la regla

La AWS IoT que está configurada para evaluar los mensajes enviados por el dispositivo y procesar los datos del dispositivo. El nombre de la regla se asignará a su destino. El destino requiere que la regla procese los mensajes que recibe. Puede elegir que los mensajes se procesen invocando un AWS IoTRegla o publicando en el AWS IoTAgente de mensajes.

- Si eligeEscriba un nombre de regla, escriba un nombre y, a continuación, elijaCopiar para copiar el nombre de la regla que introducirás al crear el AWS IoTRegla. Puede elegirCrear regla para crear la regla ahora o navegar hasta elReglasCentro de la AWS IoTconsola y cree una regla con ese nombre.

También puede escribir una regla y utilizar laAvanzadoconfiguración para especificar un nombre de tema. El nombre del tema se proporciona durante la invocación de reglas y se accede a él mediante latopicexpresión dentro de la regla. Para obtener más información acerca de las reglas del AWS IoT, consulte [Reglas para AWS IoT \(p. 472\)](#).

- Si eligePublicación enAWS IoTAgente de mensajes, introduzca un nombre de tema. A continuación, puede copiar el nombre del tema MQTT y varios suscriptores pueden suscribirse a este tema para recibir mensajes publicados en ese tema. Para obtener más información, consulte [Temas MQTT \(p. 98\)](#).

Para obtener más información acerca deAWS IoTReglas para destinos, consulte[Crear reglas para procesarLoRaMensajes de dispositivos WAN \(p. 1137\)](#).

- Role name (Nombre de rol)

El rol de IAM que concede permiso a los datos del dispositivo para tener acceso a la regla mencionada enNombre de la regla. En la consola, puede crear un nuevo rol de servicio o seleccionar uno ya existente. Si va a crear un nuevo rol de servicio, puede introducir un nombre de rol (por ejemplo,**IoTWirelessDestinationRole**), o déjelo en blanco paraAWS IoT Core for LoRaWANpara generar un nombre nuevo para el rol. AWS IoT Core for LoRaWANcreará automáticamente el rol de IAM con los permisos adecuados en su nombre.

Para obtener más información sobre los roles de IAM, consulte[Uso de roles de IAM](#).

Agregar un destino mediante la API

Si desea agregar un destino con la CLI, debe haber creado ya la regla y el rol de IAM para su destino. Para obtener más información acerca de los detalles que requiere un destino en el rol, consulte[Crear un rol de IAM para sus destinos \(p. 1136\)](#).

La lista siguiente contiene las acciones de la API que realizan las tareas asociadas con la adición, actualización o eliminación de un destino.

AWS IoT WirelessAcciones de la API para destinos

- [CreateDestination](#)
- [GetDestination](#)
- [ListDestinations](#)
- [UpdateDestination](#)
- [DeleteDestination](#)

Para ver la lista completa de acciones y tipos de datos disponibles para crear y administrarAWS IoT Core for LoRaWANrecursos, consulte la[AWS IoT WirelessReferencia de la API de](#).

Cómo para usarAWS CLIPara agregar un destino

Puede utilizar elAWS CLIPara añadir un destino mediante el[create-destination](#) comando. El siguiente ejemplo muestra cómo crear un destino mediante la introducción de un nombre de regla

mediante `RuleName` como valor de la `expression-type` parámetro. Si desea especificar un nombre de tema para publicar o suscribirse al agente de mensajes, cambie la opción `expression-type` valor del parámetro `aMqttTopicId`.

```
aws iotwireless create-destination \
--name IoTWirelessDestination \
--expression-type RuleName \
--expression IoTWirelessRule \
--role-arn arn:aws:iam::123456789012:role/IoTWirelessDestinationRole
```

Al ejecutar este comando se crea un destino con el nombre de destino, el nombre de la regla y el nombre de la función especificados. Para obtener información sobre los nombres de reglas y roles de los destinos, consulte [Crear reglas para procesar LoRaWAN Mensajes de dispositivos \(p. 1137\)](#) y [Crear un rol de IAM para sus destinos \(p. 1136\)](#).

Para obtener información sobre los CLI que puede usar, consulte [AWS CLI Referencia de de](#).

Crear un rol de IAM para sus destinos

AWS IoT Core for LoRaWAN destinos requieren roles de IAM que den AWS IoT Core for LoRaWAN los permisos necesarios para enviar datos a la AWS IoT regla. Si un rol de este tipo aún no está definido, debe definirlo para que aparezca en la lista de roles.

Cuando utilice la consola para agregar un destino, AWS IoT Core for LoRaWAN crea automáticamente un rol de IAM para usted, tal como se describe con anterioridad en este tema. Cuando agrega un destino con la API o la CLI, debe crear el rol de IAM para su destino.

Para crear una política de IAM para su AWS IoT Core for LoRaWAN rol de destino

1. Abra el icono [Centro de políticas de la consola de IAM](#).
2. Elegir Crear política, y elija la JSON Pestaña.
3. En el editor, elimine cualquier contenido del editor y pegue este documento de política.

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

4. Elegir Política de revisión, y en Nombre, introduzca un nombre para esta política. Necesitará este nombre para utilizarlo en el siguiente procedimiento.

También puede describir esta política en Descripción, si lo desea.

5. Elija Create Policy (Crear política).

Para crear un rol de IAM para un AWS IoT Core for LoRaWAN destino

1. Abra el icono [Centro de roles de la consola de IAM](#) y elige Creación de un rol.
2. En Seleccione el tipo de entidad de confianza, elige Otra Cuenta de AWS.
3. En ID de cuenta, introduzca su Cuenta de AWS ID y, a continuación, elija Siguiente: Permisos.

4. En el campo de búsqueda, escriba el nombre de la política de IAM que creó en el procedimiento anterior.
5. En los resultados de búsqueda, consulte la política de IAM que creó en el procedimiento anterior.
6. Seleccione Next (Siguiente): Etiquetas luego seleccione Siguiente: Consulte.
7. En Nombre del rol, escriba el nombre de este rol y, a continuación, elija Creación de un rol.
8. En el mensaje de confirmación, seleccione el nombre del rol que creó para editar el nuevo rol.
9. En Resumen, elige Relaciones de confianza y, a continuación, elija Modificar relación de confianza.
10. En Policy Document, cambie la Principal a similar a este ejemplo.

```
"Principal": {  
    "Service": "iotwireless.amazonaws.com"  
},
```

Después de cambiar el Principal, el documento de política completo debe tener el aspecto del siguiente ejemplo.

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

11. Para guardar los cambios y salir, elija Actualización de política de confianza.

Con este rol definido, puede encontrarlo en la lista de roles cuando configura su AWS IoT Core for LoRaWAN destinos.

Crear reglas para procesar LoRaWAN mensajes de dispositivos WAN

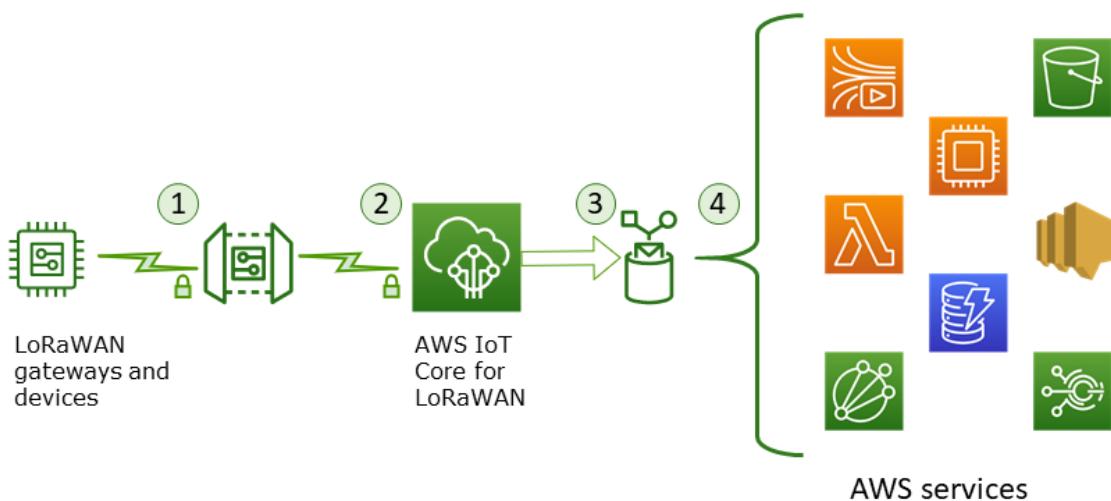
AWS IoT las reglas envían mensajes de dispositivo a otros servicios. AWS IoT las reglas también pueden procesar los mensajes binarios recibidos de un LoRaWAN dispositivo para convertir los mensajes a otros formatos que pueden facilitar su uso para otros servicios.

[AWS IoT Core for LoRaWAN destinos \(p. 1134\)](#) asocia un dispositivo inalámbrico a la regla que procesa los datos de mensajes del dispositivo para enviarlos a otros servicios. La regla actúa sobre los datos del dispositivo tan pronto como AWS IoT Core for LoRaWAN lo recibe. [AWS IoT Core for LoRaWAN destinos \(p. 1134\)](#) pueden compartirse con todos los dispositivos cuyos mensajes tienen el mismo formato de datos y que envían sus datos al mismo servicio.

Cómo AWS IoT procesa los datos de mensajes de dispositivo

Procedimiento AWS IoT la regla procesa los datos de mensajes de un dispositivo dependen del servicio que recibirá los datos, del formato de los datos del mensaje del dispositivo y del formato de datos que requiere el servicio. Normalmente, la regla llama a una AWS Lambda para convertir los datos del mensaje del dispositivo al formato que requiere un servicio y, a continuación, envía el resultado al servicio.

En la siguiente ilustración se muestra cómo se protegen y procesan los datos de los mensajes a medida que se desplazan del dispositivo inalámbrico a un AWS servicio.



1. LaLoRaEl dispositivo inalámbrico WAN cifra sus mensajes binarios utilizando el modo CTR AES128 antes de transmitirlos.
2. AWS IoT Core for LoRaWANdescifra el mensaje binario y codifica la carga útil del mensaje binario descifrada como una cadena base64.
3. El mensaje codificado en base64 resultante se envía como carga útil de mensajes binarios (una carga útil de mensajes que no tiene formato como documento JSON) alAWS IoTregla descrita en el destino asignado al dispositivo.
4. LaAWS IoTdirige los datos del mensaje al servicio descrito en la configuración de la regla.

La carga útil binaria cifrada recibida desde el dispositivo inalámbrico no se modifica ni interpreta porAWS IoT Core for LoRaWAN. La carga útil del mensaje binario descifrado se codifica solo como una cadena base64. Para que los servicios tengan acceso a los elementos de datos de la carga útil de mensajes binarios, los elementos de datos deben analizarse fuera de la carga útil mediante una función llamada por la regla. La carga útil de mensajes codificados en base64 es una cadena ASCII, por lo que podría almacenarse como tal para analizarla más adelante.

Crear reglas paraLoRaWAN

AWS IoT Core for LoRaWANutilizaAWS IoTreglas para enviar mensajes de dispositivos de forma segura directamente a otrosAWSservicios sin necesidad de utilizar el agente de mensajes. Al eliminar el agente de mensajes de la ruta de ingestión, reduce los costes y optimiza el flujo de datos.

Para unAWS IoT Core for LoRaWANregla para enviar mensajes de dispositivo a otrosAWSservicios, requiere unAWS IoT Core for LoRaWANdestino yAWS IoTregla asignada a ese destino. LaAWS IoTdebe contener una instrucción de consulta SQL y al menos una acción de regla.

Por lo general, laAWS IoTLa instrucción de consulta de regla consiste en:

- Cláusula SELECT de SQL que selecciona y da formato a los datos de la carga útil del mensaje
- Filtro de temas (el objeto FROM de la instrucción de consulta de reglas) que identifica los mensajes que se van a utilizar
- Una sentencia condicional opcional (una cláusula WHERE de SQL) que especifica las condiciones sobre las que actuar

A continuación se muestra un ejemplo de una instrucción de consulta de reglas:

```
SELECT temperature FROM iot/topic' WHERE temperature > 50
```

Cuando se construye AWS IoT Reglas para procesar cargas útiles desde LoRaWAN dispositivos, no tiene que especificar la cláusula FROM como parte del objeto de consulta de reglas. La instrucción de consulta de reglas debe tener la cláusula SELECT de SQL y, opcionalmente, puede tener la cláusula WHERE. Si la instrucción de consulta utiliza la cláusula FROM, se ignora.

A continuación se muestra un ejemplo de una declaración de consulta de reglas que puede procesar cargas útiles desde LoRaWAN dispositivos:

```
SELECT WirelessDeviceId, WirelessMetadata.LoRaWAN.FPort as FPort,  
WirelessMetadata.LoRaWAN.DevEui as DevEui,  
PayloadData
```

En este ejemplo, a la acción PayloadData es una carga binaria codificada con Base64 enviada por su LoRaWAN dispositivo.

A continuación se muestra un ejemplo de instrucción de consulta de reglas que puede realizar una decodificación binaria de la carga útil entrante y transformarla en un 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 obtener más información sobre el uso de las cláusulas SELECT AND WHERE, consulte [Referencia de la SQL de AWS IoT \(p. 562\)](#).

Para obtener más información sobre AWS IoT Reglas y cómo crearlas y utilizarlas, consulte [Reglas para AWS IoT \(p. 472\)](#) y [Crear AWS IoT Reglas para enrutar datos de dispositivos a otros servicios \(p. 193\)](#).

Para obtener información sobre cómo crear y utilizar AWS IoT Core for LoRaWAN destinos, consulte [Añadir destinos a AWS IoT Core para LoRaWAN \(p. 1134\)](#).

Para obtener información sobre el uso de cargas útiles de mensajes binarios en una regla, consulte [Uso de las cargas binarias \(p. 624\)](#).

Para obtener más información sobre la seguridad de los datos y el cifrado utilizados para proteger la carga útil de los mensajes durante su trayecto, consulte [Protección de los datos en AWS IoT Core \(p. 381\)](#).

Para obtener una arquitectura de referencia que muestra un ejemplo de decodificación binaria e implementación para reglas de IoT, consulte [AWS IoT Core for LoRaWAN Muestras de soluciones en GitHub](#).

Connect el dispositivo LoRaWAN y verifique el estado de su conexión

Para poder comprobar el estado de la conexión del dispositivo, debe haber agregado ya el dispositivo y haberlo conectado a AWS IoT Core for LoRaWAN. Para obtener información sobre cómo agregar un dispositivo, consulte [Añade tu dispositivo inalámbrico a AWS IoT Core for LoRaWAN \(p. 1130\)](#).

Después de agregar el dispositivo, consulta el manual del usuario del dispositivo para obtener información sobre cómo iniciar el envío de un mensaje de enlace ascendente desde tu LoRaWAN dispositivo.

Comprobar el estado de la conexión del dispositivo mediante la consola

Para comprobar el estado de la conexión mediante la consola, navegue hasta la [Dispositivos](#) Página de la AWS IoTconsola y elige el dispositivo que has agregado. En el navegadorDetalles desde la página de detalles de dispositivos inalámbricos, verás la fecha y la hora en que se recibió el último enlace ascendente.

Comprobar el estado de la conexión del dispositivo mediante la API

Para comprobar el estado de la conexión mediante la API, utilice `laGetWirelessDeviceStatistics` API. Esta API no tiene cuerpo de solicitud y solo contiene un cuerpo de respuesta que muestra cuándo se recibió el último vínculo ascendente.

```
HTTP/1.1 200
Content-type: application/json

{
    "LastUplinkReceivedAt": "2021-03-24T23:13:08.476015749Z",
    "LoRaWAN": {
        "DataRate": 5,
        "DevEui": "647fda000006420",
        "Frequency": 868100000
        "Gateways": [
            {
                "GatewayEui": "c0ee40ffff29df10",
                "Rssi": -67,
                "Snr": 9.75
            }
        ],
        "WirelessDeviceId": "30cbdcf3-86de-4291-bfab-5bfa2b12bad5"
    }
}
```

Pasos siguientes

Ahora que ha conectado el dispositivo y verificado el estado de la conexión, puede observar el formato de los metadatos de enlace ascendente recibidos desde el dispositivo mediante el [Cliente de prueba MQTT](#) en el [Pruebas](#) Página de la AWS IoTconsola de . Para obtener más información, consulte [Ver formato de los mensajes de enlace ascendente enviados desde LoRaWAN \(p. 1166\)](#).

Conexión aAWS IoT Core for LoRaWANmediante un punto de enlace de interfaz de la VPC

Puedes conectarte directamente aAWS IoT Core for LoRaWANa través de [Puntos de enlace de la VPC de interfaz \(AWS PrivateLink\)](#) en su Virtual Private Cloud (VPC) en lugar de conectarse a través de Internet pública. Cuando se utiliza un punto de enlace de interfaz de la VPC, la comunicación entre la VPC y AWS IoT Core for LoRaWAN se realiza en su totalidad y de manera segura dentro de la red de AWS.

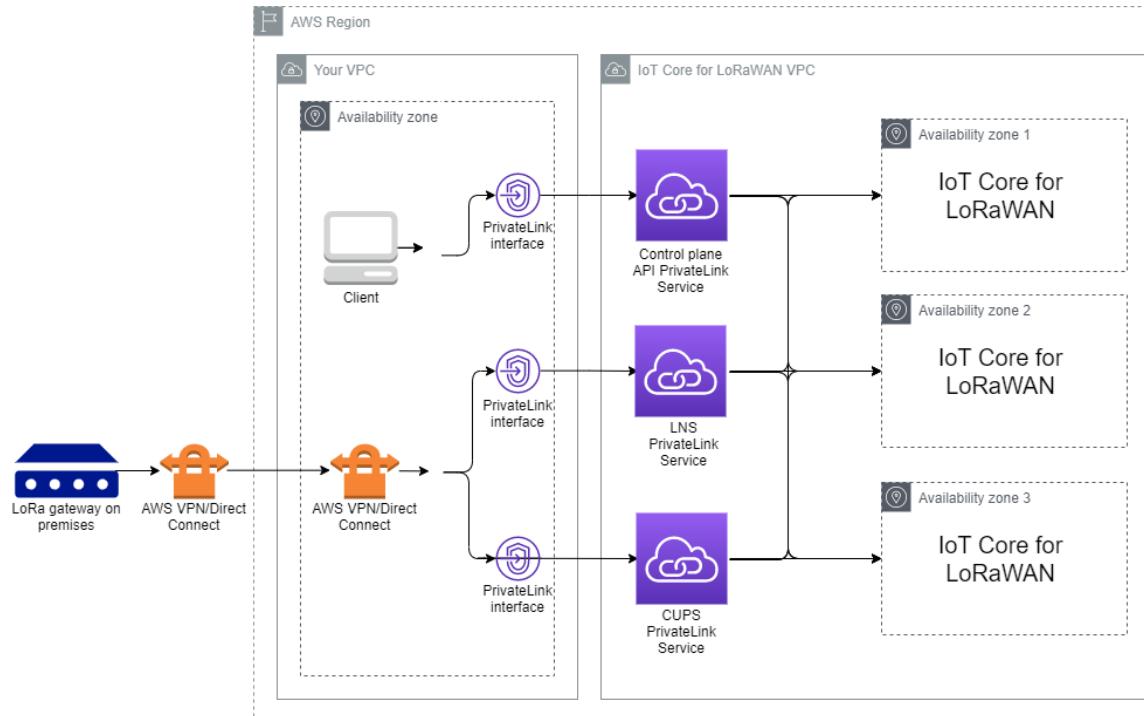
AWS IoT Core for LoRaWANadmite puntos de enlace de interfaz de Amazon Virtual Private Cloud basados en [AWS PrivateLink](#). Cada punto de enlace de la VPC está representado por una o varias Interfaces de red elásticas (ENI) con direcciones IP privadas en las subredes de la VPC.

Para obtener más información acerca de la VPC y los puntos de conexión, consulte [¿Qué es Amazon VPC?](#).

Para obtener más información acerca de [AWS PrivateLink](#), consulte [AWS PrivateLinky los puntos de conexión de la VPC](#).

AWS IoT Core for LoRaWANArquitectura de enlace privado

En el siguiente diagrama se muestra la arquitectura de enlace privado de AWS IoT Core for LoRaWAN. La arquitectura utiliza Transit Gateway y Route 53 Resolver para compartir elAWS PrivateLink endpoints de interfaz entre la VPC, elAWS IoT Core for LoRaWANVPC y un entorno local. Encontrará un diagrama de arquitectura más detallado al configurar la conexión a los extremos de la interfaz de la VPC.



Puntos de enlace de AWS IoT Core for LoRaWAN

AWS IoT Core for LoRaWANtiene tres puntos de enlace públicos. Cada endpoint público tiene un endpoint de interfaz de VPC correspondiente. Los endpoints públicos se pueden clasificar en puntos finales de plano de control y plano de datos. Para obtener más información acerca de estos puntos de enlace, consulte[AWS IoT Core for LoRaWANPuntos de enlace de API](#).

Note

AWS PrivateLinkadmite puntos de conexión solo en EE.UU. Este (Norte de Virginia) y Europa (Irlanda).

- Puntos de enlace de API del plano de control

Puede utilizar puntos de enlace de la API del plano de control para interactuar con laAWS IoT WirelessAPI. Se puede acceder a estos puntos finales desde un cliente alojado en su Amazon VPC medianteAWS PrivateLink.

- Puntos finales de API de plano de datos

Los extremos de la API del plano de datos son los endpoints de LoRaWAN Network Server (LNS) y Configuration and Update Server (CUPS) que puede utilizar para interactuar con elAWS IoT Core for LoRaWANLNS yPuntos de conexión de CUPS. Se puede acceder a estos puntos finales desde las puertas de enlace LoRa locales medianteAWS VPNoAWS Direct Connect. Obtiene estos puntos finales al incorporar su puerta de enlace aAWS IoT Core for LoRaWAN. Para obtener más información, consulte[Agregar una puerta de enlace aAWS IoT Core for LoRaWAN \(p. 1125\)](#).

Los siguientes temas muestran cómo incorporar estos puntos de conexión.

Temas

- [IncorporaciónAWS IoT Core for LoRaWANPunto de enlace de API del plano de control \(p. 1142\)](#)
- [IncorporaciónAWS IoT Core for LoRaWANPuntos de enlace de API del plano de datos \(p. 1144\)](#)

IncorporaciónAWS IoT Core for LoRaWANPunto de enlace de API del plano de control

Puede usarAWS IoT Core for LoRaWANpuntos finales de API de plano de control para interactuar con elAWS IoT WirelessAPI. Por ejemplo, puede utilizar este punto de conexión para ejecutar la[Enviar datos a un dispositivo inalámbricoAPI](#) desde la que enviar datosAWS IoT a su dispositivo LoraWan. Para obtener más información, consulte[AWS IoT Core for LoRaWANPuntos de enlace de API del plano de control](#).

Puede utilizar el cliente alojado en su Amazon VPC para acceder a los endpoints del plano de control alimentados porAWS PrivateLink. Utilizará estos puntos de enlace para conectarse a laAWS IoT WirelessAPI a través de un punto de enlace de interfaz en su Virtual Private Cloud (VPC) en lugar de conectarse a través de Internet pública.

Para incluir el extremo del plano de control:

- [Cree su Amazon VPC y subred \(p. 1142\)](#)
- [Lance una instancia de Amazon EC2 en la subred \(p. 1143\)](#)
- [Crear un punto de enlace de interfaz de Amazon VPC \(p. 1143\)](#)
- [Prueba de la conexión con el punto de enlace de interfaz \(p. 1144\)](#)

Cree su Amazon VPC y subred

Antes de poder conectarse al punto de enlace de la interfaz, debe crear una VPC y una subred. A continuación, lanzará una instancia EC2 en la subred, que puede utilizar para conectarse al extremo de la interfaz.

Para crear la VPC:

1. Vaya a la [.VPC](#)de la consola de Amazon VPC y elijaCrear una VPC.
2. En la páginaCrear una VPCpágina:
 - Escriba un nombre paraEtiqueta de nombre de VPC: opcional(por ejemplo,**VPC-A**).
 - Introduzca un rango de direcciones IPv4 para la VPC en el bloque CIDR IPv4 (por ejemplo,**10.100.0.0/16**).
3. Mantenga los valores predeterminados para otros campos y elijaCrear una VPC.

Para crear tu subred:

1. Vaya a la [.Subredes](#)de la consola de Amazon VPC y elijaCrear subred.
2. En la páginaCrear subredpágina:
 - ParaID DE LA VPC, elija la VPC que creó anteriormente (por ejemplo,**VPC-A**).
 - Escriba un nombre paraSubnet name (Nombre de la subred)(por ejemplo,**Private subnet**).
 - Elija el iconoZona de disponibilidadpara su subred.
 - Introduzca el bloque de direcciones IP de su subred en ellIPv4 CIDR blocken formato CIDR (por ejemplo,**10.100.0.0/24**).
3. Para crear la subred y agregarla a la VPC, elijaCrear subred.

Para obtener más información, consulte[Trabajar con VPC y subredes](#).

Lance una instancia de Amazon EC2 en la subred

Para lanzar una instancia EC2:

1. Vaya a la [Amazon EC2](#)Consola y elijaLanzar instancia.
2. Para AMI, elijaAmazon Linux 2 AMI (HVM), SSD Volume Typey, a continuación, elija lat2 microtipo de instancia. Para configurar los detalles de la instancia, elijaPróximo.
3. En el navegadorPágina Configure Instance Details (Configurar los detalles de la instancia)página:
 - ParaRed, elija la VPC que creó anteriormente (por ejemplo,VPC-A).
 - ParaSubred, elija la subred que creó anteriormente (por ejemplo,**Private subnet**).
 - ParaRol de IAM, elija el rolAWSIoTWirelessFullAccessconcederAWS IoT Core for LoRaWANpolítica de acceso completo. Para obtener más información, consulte[AWSIoTWirelessFullAccessResumen de políticas](#).
 - ParaSupongamos una IP privada, utilice una dirección IP, por ejemplo,10.100.0.42.
4. Seleccione Next (Siguiente): Adición de almacenamientoy luego elijaSiguiente: Añadir etiquetas. Opcionalmente, puede agregar cualquier etiqueta para asociarla a la instancia de EC2. Seleccione Next (Siguiente): Configure Security Group (Configurar grupo de seguridad).
5. En el navegadorPágina Configure Security Group (Configurar grupo de seguridad), configure el grupo de seguridad para permitir:
 - AbiertoAll TCPpara Source as10.200.0.0/16.
 - AbiertoTodo el ICMP - IPV4para Source as10.200.0.0/16.
6. Para revisar los detalles de la instancia y lanzar la instancia EC2, elijaRevisar y lanzar.

Para obtener más información, consulte[Introducción a las instancias Linux de Amazon EC2](#).

Crear un punto de enlace de interfaz de Amazon VPC

Puede crear un punto de enlace de la VPC para su VPC, a la que puede acceder a ella la API de EC2.
Para crear el punto de enlace:

1. Vaya a la [VPC](#): Puntos de enlace deconsola de y elijaCrearPunto de enlace.
2. En el navegadorCreación de un punto de enlace, especifique la información siguiente.
 - ElegirServicio de AWSsparaCategoría de servicio.
 - ParaNombre del servicio, busque introduciendo la palabra clave*iotwireless*. En la lista de*iotwireless*servicios mostrados, elija el extremo de la API del plano de control para su región. El punto final tendrá el formato`com.amazonaws.region.iotwireless.api`.
 - ParaVPC:ySubredes, elija la VPC en la que desee crear el punto de conexión y las zonas de disponibilidad (AZ) en las que desee crear la red de punto de conexión.

Note

La*iotwireless*puede que el servicio no sea compatible con todas las zonas de disponibilidad.

- ParaHabilitar nombre de DNS, eligeHabilitar para este endpoint.

Si elige esta opción, se resolverá automáticamente el DNS y creará una ruta enAmazon Route 53 Public Data Planede modo que las API que utilice más adelante para probar la conexión pasen por los endpoints privatelink.

- ParaGrupo de seguridad, elija los grupos de seguridad que desee asociarse a las interfaces de red de punto de conexión.
- Opcionalmente, puede agregar o eliminar etiquetas. Las etiquetas son pares nombre-valor que utiliza para asociarse a su punto de conexión.

3. Para crear el endpoint de la VPC, elijaCrear punto de conexión.

Prueba de la conexión con el punto de enlace de interfaz

Puede utilizar un SSH para acceder a su instancia de Amazon EC2 y, a continuación, utilizar el AWS CLI para conectarse a los endpoints de la interfaz privatelink.

Antes de conectarse al punto final de la interfaz, descargue el más reciente AWS CLI versión siguiendo las instrucciones descritas en [Instalación, actualización y desinstalación AWS CLI versión 2 en Linux](#).

Los siguientes ejemplos muestran cómo puede probar la conexión con el punto de enlace de interfaz a través de la CLI.

```
aws iotwireless create-service-profile \  
  --endpoint-url https://api.iotwireless.region.amazonaws.com \  
  --name='test-privatelink'
```

El siguiente es un ejemplo de cómo ejecutar el comando.

```
Response:  
{  
  "Arn": "arn:aws:iotwireless:region:acct_number:ServiceProfile/1a2345ba-4c5d-67b0-ab67-e0c8342f2857",  
  "Id": "1a2345ba-4c5d-67b0-ab67-e0c8342f2857"  
}
```

Del mismo modo, puede ejecutar los siguientes comandos para obtener la información del perfil de servicio o enumerar todos los perfiles de servicio.

```
aws iotwireless get-service-profile \  
  --endpoint-url https://api.iotwireless.region.amazonaws.com  
  --id="1a2345ba-4c5d-67b0-ab67-e0c8342f2857"
```

A continuación se muestra un ejemplo del comando list-device-profiles.

```
aws iotwireless list-device-profiles \  
  --endpoint-url https://api.iotwireless.region.amazonaws.com
```

IncorporaciónAWS IoT Core for LoRaWANPuntos de enlace de API del plano de datos

AWS IoT Core for LoRaWAN los puntos finales del plano de datos constan de los siguientes extremos. Obtiene estos puntos finales al agregar la puerta de enlace a AWS IoT Core for LoRaWAN. Para obtener más información, consulte [Agregar una puerta de enlace a AWS IoT Core for LoRaWAN \(p. 1125\)](#).

- Endpoints de LoRaWAN Network Server (LNS)

Los extremos de LNS tienen el formato **account-specific-prefix.lns.lorawan.**region**.amazonaws.com**. Puede utilizar este punto final para establecer una conexión para intercambiar mensajes de enlace ascendente y descendente de LoRa.

- Endpoints del servidor de configuración y actualización (CUPS)

Los puntos finales de CUPS tienen el formato **account-specific-prefix.cups.lorawan.**region**.amazonaws.com**. Puede utilizar este endpoint para la

administración de credenciales, la configuración remota y la actualización de firmware de las puertas de enlace.

Para obtener más información, consulte [Uso de protocolos CUPS y LNS \(p. 1151\)](#).

Para buscar los puntos finales de la API de Data Plane para suCuenta de AWSy Region, utilice la[get-service-endpoint](#)El comando CLI que se muestra aquí, o la[GetServiceEndpoint](#)API DE DESCANSO. Para obtener más información, consulte[AWS IoT Core for LoRaWANPuntos de enlace de API de plano de datos](#).

Puede conectar su puerta de enlace LoraWAN en las instalaciones para comunicarse conAWS IoT Core for LoRaWANPuntos de enlace de . Para establecer esta conexión, primero conecte la puerta de enlace local a suCuenta de AWSen la VPC mediante una conexión VPN. A continuación, puede comunicarse con los puntos finales de la interfaz del plano de datos en elAWS IoT Core for LoRaWANVPC que funciona con privatelink.

A continuación se muestra cómo incorporar estos puntos de conexión.

- [Crear endpoint de interfaz de VPC y zona alojada privada \(p. 1145\)](#)
- [Usar VPN para conectar puertas de enlace LoRa a suCuenta de AWS \(p. 1148\)](#)

Crear endpoint de interfaz de VPC y zona alojada privada

AWS IoT Core for LoRaWANtiene dos puntos finales de plano de datos, endpoint del servidor de configuración y actualización (CUPS) y endpoint de LoRaWAN Network Server (LNS). El proceso de configuración para establecer una conexión de enlace privado a ambos extremos es el mismo, por lo que podemos utilizar el endpoint LNS para fines ilustrativos.

Para los endpoints de su plano de datos, las puertas de enlace LoRa se conectan primero a suCuenta de AWSen Amazon VPC, que luego se conecta al punto final de la VPC en elAWS IoT Core for LoRaWANVPC.

Al conectarse a los endpoints, los nombres DNS se pueden resolver en una VPC pero no se pueden resolver en varias VPC. Para deshabilitar el DNS privado al crear el endpoint, deshabilite elHabilitar nombre de DNSconfiguración de. Puede utilizar la zona hospedada privada para proporcionar información sobre cómo desea que Route 53 responda a las consultas de DNS de sus VPC. Para compartir su VPC con un entorno local, puede utilizar unRoute 53 Resolver tpara facilitar el DNS híbrido.

Para completar este procedimiento, siga estos pasos.

- [Creación de una subred y una Amazon VPC \(p. 1145\)](#)
- [Crear un punto de enlace de interfaz de Amazon VPC \(p. 1145\)](#)
- [Configurar una zona alojada privada \(p. 1146\)](#)
- [Configurar el solucionador entrante Route 53 \(p. 1147\)](#)
- [Pasos siguientes \(p. 1148\)](#)

Creación de una subred y una Amazon VPC

Puede reutilizar la Amazon VPC y la subred que creó al incorporar el endpoint del plano de control. Para obtener información, consulte [Cree su Amazon VPC y subred \(p. 1142\)](#).

Crear un punto de enlace de interfaz de Amazon VPC

Puede crear un extremo de VPC para la VPC, que es similar a cómo crearía uno para el extremo del plano de control.

1. Vaya a la [.VPC](#): Puntos de enlace de consola de y elija Creación de un punto de enlace.
2. En el navegador Creación de un punto de enlace, especifique la información siguiente.
 - Elegir Servicio de AWS para Categoría de servicio.
 - Para Nombre del servicio, busque introduciendo la palabra clave **lns**. En la lista de los servicios mostrados, elija el extremo de la API del plano de datos LNS para su región. El punto final tendrá el formato `com.amazonaws.region.lorawan.lns`.

Note

Si sigue este procedimiento para el endpoint CUPS, busque `cups`. El punto final tendrá el formato `com.amazonaws.region.lorawan.cups`.

- Para VPC y Subredes, elija la VPC en la que deseé crear el punto de conexión y las zonas de disponibilidad (AZ) en las que deseé crear la red de punto de conexión.

Note

La IoTWireless puede que el servicio no sea compatible con todas las zonas de disponibilidad.

- Para Habilitar nombre de DNS, asegúrese de que Habilitar para este endpoint no está seleccionado.

Si no selecciona esta opción, puede deshabilitar el DNS privado para el endpoint de la VPC y utilizar la zona alojada privada en su lugar.

- Para Grupo de seguridad, elija los grupos de seguridad que deseé asociarse a las interfaces de red de punto de conexión.
- Opcionalmente, puede agregar o eliminar etiquetas. Las etiquetas son pares nombre-valor que utiliza para asociarse a su punto de conexión.

3. Para crear el endpoint de la VPC, elija Crear un punto de enlace.

Configurar una zona alojada privada

Después de crear el endpoint de enlace privatelink, en el Detalles de su punto de conexión, verá una lista de nombres de DNS. Puede utilizar uno de estos nombres DNS para configurar la zona alojada privada. El nombre DNS tendrá el formato `vpce-xxxx.lns.lorawan.region.vpce.amazonaws.com`.

Crear la zona alojada privada

Para crear la zona alojada privada:

1. Vaya a la [Route 53 Zonas hospedadas](#) consola de y elija Crear zona alojada.
2. En el navegador Crear zona alojada, especifique la información siguiente.
 - Para Domain name (Nombre del dominio), introduzca el nombre del servicio completo del endpoint `LNS.lns.lorawan.region.amazonaws.com`.

Note

Si está siguiendo este procedimiento para el endpoint CUPS, introduzca `cups.lorawan.region.amazonaws.com`.

- Para Tipo, elige Zona hospedada privada.
 - Opcionalmente, puedes agregar o quitar etiquetas para asociarlas a tu zona alojada.
3. Para crear tu zona alojada privada, elige Crear zona alojada.

Para obtener más información, consulte [Creación de una zona alojada privada](#).

Después de crear una zona alojada privada, puede crear un registro que indique al DNS cómo desea que el tráfico se enrute a ese dominio.

Creación de un registro

Después de crear una zona alojada privada, puede crear un registro que indique al DNS cómo desea que el tráfico se enrute a ese dominio. Para crear un registro:

1. En la lista de zonas hospedadas que se muestra, elija la zona hospedada privada que creó anteriormente y elijaCrear registro.
2. Utilice el método del asistente para crear el registro. Si la consola te presenta elCreación rápida, elijaCambiar al asistente.
3. ElegirDireccionamiento simpleparaPolítica de direccionamiento y luego elijaPróximo.
4. En el navegadorConfigurar registros, elijaDefina registro simple.
5. En el navegadorDefina registro simplepágina:
 - ParaNombre del registro, introduzca el alias de suCuenta de AWSnúmero. Obtiene este valor al incorporar su puerta de enlace o mediante el[GetServiceEndpointAPI DE DESCANSO](#).
 - ParaTipo de registro, mantenga el valor comoA – Routes traffic to an IPv4 address and some AWS resources.
 - ParaValor/ruta de destino del tráfico, eligeAlias a punto de enlace de la VPC. A continuación, elija suRegióny, a continuación, elija el punto de conexión de que creó anteriormente, como se describe en[Crear un punto de enlace de interfaz de Amazon VPC \(p. 1145\)](#)de la lista de puntos finales que se muestra.
6. ElegirDefina registro simplepara crear tu registro.

Configurar el solucionador entrante Route 53

Para compartir un endpoint de VPC en un entorno local, se puede utilizar un solucionador Route 53 para facilitar el DNS híbrido. El solucionador de entrada le permitirá enrutar el tráfico desde la red local a los endpoints del plano de datos sin pasar por Internet pública. Para devolver los valores de dirección IP privada del servicio, cree Route 53 Resolver en la misma VPC que el extremo de la VPC.

Cuando crea el solucionador entrante, solo tiene que especificar la VPC y las subredes que creó anteriormente en las zonas de disponibilidad (AZ). Route 53 Resolver utiliza esta información para asignar automáticamente una dirección IP para enrutar el tráfico a cada una de las subredes.

Para crear el solucionador entrante:

1. Vaya a la [.Route 53 Puntos de enlace de entrada](#)consola de y elijaCrear un punto de enlace de entrada.

Note

Asegúrese de utilizar lo mismoRegion de AWSque utilizó al crear el endpoint y la zona alojada privada.

2. En el navegadorCrear un punto de enlace de entrada, especifique la información siguiente.
 - Escriba un nombre paraEndpoint name (Nombre del punto de enlace)(por ejemplo,**VPC_A_Test**).
 - ParaVPC in the region, elija la misma VPC que utilizó al crear el endpoint de la VPC.
 - Configuración delSecurity group for this endpoint (Grupo de seguridad para este punto de enlace)para permitir el tráfico entrante de la red local.
 - Para Dirección IP, elijaUtilice una dirección IP que se seleccione automáticamente.
3. ElegirEnviarpara crear tu solucionador entrante.

Por ejemplo, supongamos que las direcciones IP10.100.0.145y10.100.192.10se asignaron para el Resolver Route 53 entrante para el tráfico de enrutamiento.

Pasos siguientes

Ha creado la zona alojada privada y un solucionador entrante para enrutar el tráfico de sus entradas DNS. Ahora puede utilizar una VPN de Site-to-Site VPN o un punto de enlace de Client VPN. Para obtener más información, consulte [Usar VPN para conectar puertas de enlace LoRa a suCuenta de AWS \(p. 1148\)](#).

Usar VPN para conectar puertas de enlace LoRa a suCuenta de AWS

Para conectar las puertas de enlace en las instalaciones a suCuenta de AWS, puede utilizar una conexión de Site-to-Site VPN o un punto de enlace de Client VPN.

Antes de poder conectar las puertas de enlace locales, debe haber creado el punto final de la VPC y haber configurado una zona alojada privada y un solucionador de entrada para que el tráfico de las puertas de enlace no pase a través de Internet pública. Para obtener más información, consulte [Crear endpoint de interfaz de VPC y zona alojada privada \(p. 1145\)](#).

Punto de enlace de Site-to-Site VPN

Si no tiene el hardware de la puerta de enlace o desea probar la conexión VPN utilizando otroCuenta de AWS, puede utilizar una conexión de Site-to-Site VPN. Puede utilizar la Site-to-Site VPN para conectarse a los puntos de enlace de la VPC desde el mismoCuenta de AWSu otroCuenta de AWSque podrías estar usando en otroRegión de AWS.

Note

Si tienes el hardware de la puerta de enlace contigo y quieres configurar una conexión VPN, te recomendamos que uses Client VPN en su lugar. Para obtener instrucciones, consulte [Punto de enlace de Client VPN \(p. 1149\)](#).

Para configurar una Site-to-Site VPN:

1. Cree otra VPC en el sitio desde el que desea configurar la conexión. ParaVPC-A, puede volver a utilizar la VPC que creó anteriormente. Para crear otra VPC (por ejemplo,VPC-B), utilice un bloque de CIDR que no se superponga al bloque CIDR de la VPC que creó anteriormente.

Para obtener más información acerca de la configuración de las VPC, siga las instrucciones descritas en[AWSconfigurar una conexión de Site-to-Site VPN](#).

Note

El método Site-to-Site VPN descrito en el documento utiliza OpenSwan para la conexión VPN, que solo admite un túnel VPN. Si utiliza otro software comercial para la VPN, es posible que pueda configurar dos túneles entre los sitios.

2. Después de configurar la conexión de VPN, actualice la/etc/resolv.confañadiendo la dirección IP del solucionador entrante desde suCuenta de AWS. Utilizará esta dirección IP para el servidor de nombres. Para obtener más información acerca de cómo obtener esta dirección IP, consulte[Configurar el solucionador entrante Route 53 \(p. 1147\)](#). En este ejemplo, podemos utilizar la dirección IP10.100.0.145que se asignó cuando creó el Solucionador Route 53.

```
options timeout:2 attempts:5
; generated by /usr/sbin/dhclient-script
search region.compute.internal
nameserver 10.100.0.145
```

3. Ahora podemos probar si la conexión de VPN utiliza laAWS PrivateLinkpunto de conexión en lugar de pasar a través de la red de Internet pública utilizando unnslookupcomando. El siguiente es un ejemplo de cómo ejecutar el comando.

```
nslookup account-specific-prefix.lns.lorawan.region.amazonaws.com
```

A continuación se muestra un ejemplo de resultado de la ejecución del comando, que muestra una dirección IP privada que indica que la conexión se ha establecido con el AWS PrivateLink Punto de enlace de LNS.

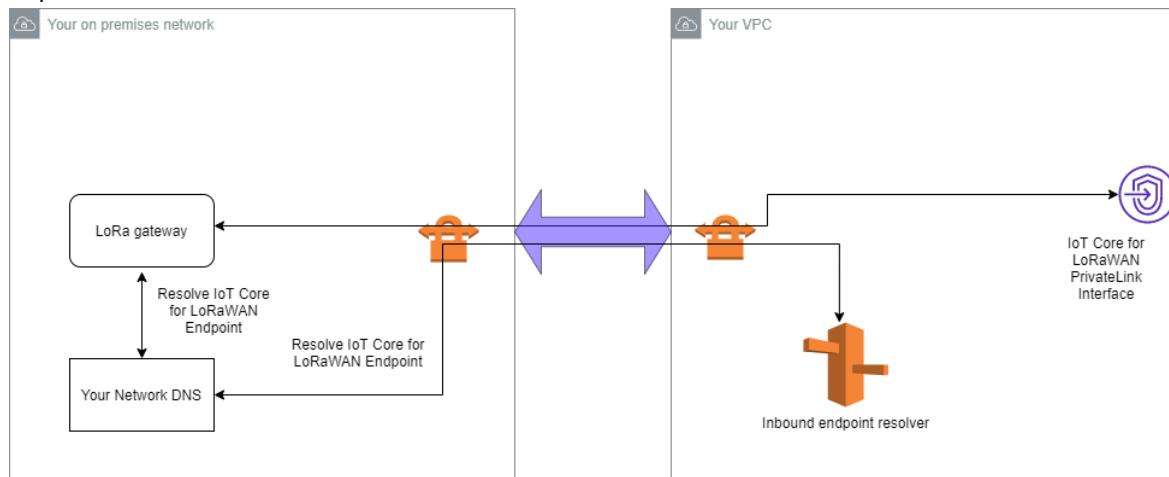
```
Server: 10.100.0.145
Address: 10.100.0.145

Non-authoritative answer:
Name: https://xxxxx.lns.lorawan.region.amazonaws.com
Address: 10.100.0.204
```

Para obtener más información acerca de cómo utilizar una conexión de Site-to-Site VPN, consulte [Funcionamiento de Site-to-Site VPN](#).

Punto de enlace de Client VPN

AWS Client VPN es un servicio de VPN basado en cliente administrado que le permite obtener acceso de forma segura a sus recursos de AWS y a los recursos de la red local. A continuación se muestra la arquitectura del servicio VPN cliente.



Para establecer una conexión de VPN a un punto de enlace de Client VPN:

1. Cree un punto de enlace de Client VPN siguiendo las instrucciones descritas en [Introducción al AWS Client VPN](#).
2. Inicie sesión en su red local (por ejemplo, un enrutador Wi-Fi) utilizando la URL de acceso de ese router (por ejemplo, 192.168.1.1), y encuentre el nombre de raíz y la contraseña.
3. Configure la puerta de enlace LoraWAN siguiendo las instrucciones de la documentación de la puerta de enlace y, a continuación, agregue la puerta de enlace a AWS IoT Core for LoRaWAN. Para obtener más información acerca de cómo agregar la gateway, consulte [Incorporación de sus gateways a AWS IoT Core for LoRaWAN \(p. 1123\)](#).
4. Compruebe si el firmware de su gateway está actualizado. Si el firmware está desactualizado, puede seguir las instrucciones proporcionadas en la red local para actualizar el firmware de la puerta de enlace. Para obtener más información, consulte [Actualizar el firmware de la puerta de enlace mediante el servicio AWS IoT Core for LoRaWAN \(p. 1154\)](#).

5. Verifique si OpenVPN se ha habilitado. Si se ha habilitado, vaya al siguiente paso para configurar el cliente OpenVPN dentro de la red local. Si no se ha habilitado, siga las instrucciones de [Guía para instalar OpenVPN para OpenWrt](#).

Note

En este ejemplo, usamos OpenVPN. Puede utilizar otros clientes VPN, comoAWS VPNoAWS Direct ConnectPara configurar la conexión de Client VPN.

6. Configure el cliente OpenVPN en función de la información de la configuración del cliente y cómo puede usar[Cliente OpenVPN mediante LUCi](#).
7. SSH a la red local y actualice la /etc/resolv.confañadiendo la dirección IP del solucionador entrante en suCuenta de AWS(10.100.0.145).
8. Para que el tráfico de la puerta de enlace utiliceAWS PrivateLinkpara conectarse al endpoint, sustituya la primera entrada DNS de la puerta de enlace a la dirección IP del solucionador entrante.

Para obtener más información acerca de cómo utilizar una conexión de Site-to-Site VPN, consulte[Introducción a Client VPN](#).

Connect a terminales de VPC de LNS y CUPS

A continuación se muestra cómo puede probar la conexión a los extremos de la VPC de LNS y CUPS.

Punto de enlace de CUPS de

Para probar laAWS PrivateLinkconexión al punto de enlace de CUPS desde su gateway de LoRa, ejecute el comando siguiente:

```
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": "xxxxxxxxxxxxxx",
        "cupsUri": "https://xxxx.cups.lorawan.region.amazonaws.com:443",
        "cupsCredCrc":1234, "tcCredCrc":552384314
      }'
      -output cups.out
```

Punto de enlace de LNS de prueba

Para probar su endpoint LNS, aprovisione primero un dispositivo LoraWAN que funcionará con la puerta de enlace inalámbrica. A continuación, puede agregar el dispositivo y ejecutar launirsetras el cual puede comenzar a enviar mensajes de enlace ascendente.

Administración de gateways conAWS IoT Corefor LoRaWAN

Las puertas de enlace actúan como puente y transportan datos de dispositivos LoRaWAN hacia y desde un servidor de red, generalmente a través de redes de gran ancho de banda como Wi-Fi, Ethernet o celular. Las puertas de enlace LoRaWAN conectan dispositivos inalámbricos aAWS IoT Core for LoRaWAN.

A continuación se presentan algunas consideraciones importantes al utilizar las puertas de enlace conAWS IoT Core for LoRaWAN. Para obtener información sobre cómo agregar la gateway aAWS IoT Core for LoRaWAN, consulte[Incorporación de sus gateways aAWS IoT Core for LoRaWAN \(p. 1123\)](#).

Requisito de software LoRa Basics Station

Para conectar a AWS IoT Core for LoRaWAN, su puerta de enlace LoRaWAN debe tener un software llamado [LoRA Basics Station](#) corriendo en él. LoRa Basics Station es un software de código abierto que Semtech Corporation mantiene y distribuido por su [GitHub](#). AWS IoT Core for LoRaWAN admite LoRa Basics Station versión 2.0.4 y posteriores.

Uso de puertas de enlace calificadas desde el AWS Catálogo de dispositivos asociados

La [AWS Catálogo de dispositivos asociados](#) contiene gateways y kits de desarrollador que están calificados para su uso con AWS IoT Core for LoRaWAN. Le recomendamos que utilice estas puertas de enlace calificadas porque no tiene que modificar el software de incrustación para conectar las puertas de enlace a AWS IoT Core. Estas puertas de enlace ya tienen una versión del software BasicStation compatible con AWS IoT Core for LoRaWAN.

Note

Si tiene una puerta de enlace que no figura en el catálogo de socios como puerta de enlace calificada con AWS IoT Core for LoRaWAN, es posible que aún pueda utilizarlo si la puerta de enlace ejecuta el software LoRa Basics Station con la versión 2.0.4 y posterior. Asegúrese de usar Autenticación de cliente y servidor TLS para autenticar su puerta de enlace LoRaWAN.

Uso de protocolos CUPS y LNS

El software LoRaWan Basics Station contiene dos subprotocolos para conectar puertas de enlace a servidores de red, LoRaWan Network Server (LNS) y Configuration and Update Server (CUPS).

El protocolo LNS establece una conexión de datos entre una puerta de enlace compatible con LoRa Basics Station y un servidor de red. Los mensajes de enlace ascendente y descendente de LoRa se intercambian a través de esta conexión de datos a través de WebSockets seguros.

El protocolo CUPS permite la administración de credenciales y la configuración remota y la actualización de firmware de las puertas de enlace. AWS IoT Core for LoRaWAN proporciona puntos finales LNS y CUPS para la ingestión de datos de LoRaWAN y la administración remota de puertas de enlace respectivamente.

Para obtener más información, consulte [Protocolo LNS](#) y [Protocolo CUPS](#).

Configurar las subbandas y las capacidades de filtrado de la puerta de enlace

Las puertas de enlace LoRaWan ejecutan un [LoRA Basics Station](#) software que permite que las puertas de enlace se conecten a AWS IoT Core for LoRaWAN. Para conectar a AWS IoT Core for LoRaWAN, la puerta de enlace LoRa consulta primero al servidor CUPS para el endpoint LNS y, a continuación, establece una conexión de datos de WebSockets con ese endpoint. Una vez establecida la conexión, los marcos de enlace ascendente y descendente pueden intercambiarse a través de esa conexión.

Filtrado de marcos de datos de LoRa recibidos por gateway

Después de que la puerta de enlace LoraWAN establezca una conexión con el endpoint, AWS IoT Core for LoRaWAN responde con un `router_config` mensaje que especifica un conjunto de parámetros para la configuración de la puerta de enlace LoRa, incluidos los parámetros de filtrado `NetID` y `JoinEui`. Para

obtener más información acerca de `router_config` cómo se establece una conexión con el servidor de red LoRaWAN (LNS), consulte [Protocolo 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  
}
```

Las puertas de enlace transportan datos de dispositivos LoRaWAN hacia y desde LNS, por lo general, a través de redes de gran ancho de banda como Wi-Fi, Ethernet o Cellular. Por lo general, las puertas de enlace recogen todos los mensajes y atraviesan el tráfico que llega a AWS IoT Core for LoRaWAN. Sin embargo, puede configurar las puertas de enlace para filtrar parte del tráfico de datos del dispositivo, lo que ayuda a conservar el uso del ancho de banda y reduce el flujo de tráfico entre la puerta de enlace y el LNS.

Para configurar la puerta de enlace LoRa para filtrar los marcos de datos, puede utilizar los parámetros `NetID`, `JoinEui` en la `router_config` message. `NetID` es una lista de valores `NetID` aceptados. Se eliminará cualquier marco de datos de LoRa que contenga un marco de datos distinto de los enumerados. `JoinEui` es una lista de pares de valores enteros que codifican rangos de valores `JoinEUI`. La puerta de enlace eliminará tramas de solicitud de unión, a menos que el campo `JoinEui` en el mensaje está dentro del rango [`begeUI`, `EndeUI`].

Canales de frecuencia y subbandas

Para las regiones RF US915 y AU915, los dispositivos inalámbricos tienen opciones de 64 canales de enlace ascendente de 125 kHz y 8 500 kHz para acceder a las redes LoRaWAN mediante las puertas de enlace LoRa. Los canales de frecuencia de enlace ascendente se dividen en 8 subbandas, cada una con 8 canales de 125 kHz y un canal de 500 kHz. Para cada puerta de enlace normal de la región AU915, se admitirán una o más subbandas.

Algunos dispositivos inalámbricos no pueden saltar entre subbandas y utilizar los canales de frecuencia en una sola subbanda cuando se conectan a AWS IoT Core for LoRaWAN. Para que se transmitan los paquetes de enlace ascendente de esos dispositivos, configure las puertas de enlace LoRa para que utilicen esa subbanda en particular. Para puertas de enlace de otras regiones de RF, como EU868, esta configuración no es necesaria.

Configure la gateway para utilizar filtrado y subbandas mediante la consola

Puede configurar la puerta de enlace para utilizar una subbanda determinada y también habilitar la capacidad de filtrar los marcos de datos de LoRa. Para especificar estos parámetros mediante la consola:

1. Vaya a la [AWS IoT Core for LoRaWAN Gateways](#) de la Página de la AWS IoT consola y elija Adición de gateway.
2. Especifique los detalles de la puerta de enlace, como la EUI de Gateway, Banda de frecuencia (RFRegion) y un opcional Nombre y Descripción y elija si desea asociar un AWS IoT cosa de su gateway. Para obtener más información acerca de cómo agregar una gateway, consulte [Añada una gateway con la consola \(p. 1125\)](#).
3. En el navegador Configuración LoRaWAN, puede especificar las subbandas y la información de filtrado.

- **SubBands:** Para añadir una subbanda, elijaAdición de subbanday especifique una lista de valores enteros que indican qué subbandas admite la puerta de enlace. LaSubBandssolo se puede configurar enRfRegionUS915 y AU915 y deben tener valores en el rango[1 , 8]dentro de una de estas regiones admitidas.
 - **NetIdFilters:** Para filtrar marcos de enlace ascendente, elijaAdición de NetIdy especifique una lista de valores de cadena que utiliza la puerta de enlace. El NetID del marco de enlace ascendente entrante del dispositivo inalámbrico debe coincidir al menos con uno de los valores enumerados; de lo contrario, el fotograma se eliminará.
 - **JoinEuiFilters:** ElijaAdición de un rango JoinEuiy especifique una lista de pares de valores de cadena que utiliza una puerta de enlace para filtrar tramas LoRa. El valor JoinEui especificado como parte de la solicitud de unión del dispositivo inalámbrico debe estar dentro del rango de al menos uno de los valores de JoinEuirange, cada uno de los que aparece como un par de [begeUI, EndeUI]; de lo contrario, el marco se eliminará.
4. A continuación, puede continuar configurando la puerta de enlace siguiendo las instrucciones descritas en[Añada una gateway con la consola \(p. 1125\)](#).

Después de agregar una puerta de enlace, en el[AWS IoT Core for LoRaWAN Gateways](#) dePágina de laAWS IoT, si seleccionas la puerta de enlace que has agregado, puedes ver laSubBandsy filtrosNetIdFiltersyJoinEuiFiltersen laDetalles específicos de LoRaWANde la página de detalles del gateway.

Configure la gateway para utilizar filtrado y subbandas mediante la API

Puede utilizar el[Crear puerta de enlace inalámbrica](#)API que utiliza para crear una puerta de enlace para configurar las subbandas que desea utilizar y habilitar la capacidad de filtrado. Uso deCreateWirelessGatewayAPI, puede especificar las subbandas y los filtros como parte de la información de configuración de la puerta de enlace que proporciona mediante elLoRAWAN. A continuación se muestra el token de solicitud que incluye esta información.

```
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": ["00000000", "00000001"],
    "RfRegion": "US915",
    "SubBands": [2]
  },
  "Name": "myFirstLoRaWANGateway",
  "ThingArn": null,
  "ThingName": null
}
```

También puede utilizar la[Actualizar puerta de enlace inalámbrica](#)API para actualizar los filtros pero no las subbandas. Si el archivo deJoinEuiFiltersyNetIdfilterslos valores son nulos, significa que no hay actualización para los campos. Si los valores no son nulos y se incluyen listas vacías, se aplica la actualización. Para obtener los valores de los campos que ha especificado, utilice la[Obtener puerta de enlace inalámbrica](#)API.

Actualizar el firmware de la puerta de enlace mediante el servicioAWS IoT Core for LoRaWAN

LaLoRA Basics Stationel software que se ejecuta en la puerta de enlace proporciona administración de credenciales e interfaz de actualización de firmware mediante el protocolo Servidor de configuración y actualización (CUPS). El protocolo CUPS proporciona una entrega segura de actualizaciones de firmware con firmas ECDSA.

Tendrás que actualizar con frecuencia el firmware de tu puerta de enlace. Puede utilizar el servicio CUPS conAWS IoT Core for LoRaWANpara proporcionar actualizaciones de firmware a la puerta de enlace donde también se pueden firmar las actualizaciones. Para actualizar el firmware de la puerta de enlace, puede utilizar el SDK o la CLI pero no la consola.

El proceso de actualización tarda 45 minutos en completarse. Puede tardar más si configura la puerta de enlace por primera vez en conectarse aAWS IoT Core for LoRaWAN. Los fabricantes de puertas de enlace suelen proporcionar sus propios archivos de actualización de firmware y firmas para que pueda utilizarlo y continuar[Cargue el archivo de firmware en un bucket de S3 y agregue una función de IAM \(p. 1157\)](#).

Si no dispone de los archivos de actualización del firmware, consulte[Generar el archivo de actualización de firmware y la firma \(p. 1154\)](#)por ejemplo que puedes utilizar para adaptarte a tu aplicación.

Para realizar la actualización del firmware de la puerta de enlace:

- [Generar el archivo de actualización de firmware y la firma \(p. 1154\)](#)
- [Cargue el archivo de firmware en un bucket de S3 y agregue una función de IAM \(p. 1157\)](#)
- [Programar y ejecutar la actualización del firmware mediante una definición de tarea \(p. 1160\)](#)

Generar el archivo de actualización de firmware y la firma

Los pasos en este procedimiento son opcionales y dependen de la puerta de enlace que utilice. Los fabricantes de puertas de enlace proporcionan su propia actualización de firmware en forma de archivo de actualización o script y Basics Station ejecuta este script en segundo plano. En este caso, lo más probable es que encuentres el archivo de actualización de firmware en las notas de la versión de la puerta de enlace que estás utilizando. A continuación, puede utilizar ese archivo o script de actualización en su lugar y proceder a[Cargue el archivo de firmware en un bucket de S3 y agregue una función de IAM \(p. 1157\)](#).

Si no tiene este script, a continuación se muestran los comandos que se deben ejecutar para generar el archivo de actualización de firmware. Las actualizaciones también se pueden firmar para garantizar que el código no se haya alterado o dañado y que los dispositivos ejecuten código publicado solo por autores de confianza.

En este procedimiento, hará lo siguiente:

- [Generar el archivo de actualización de firmware \(p. 1154\)](#)
- [Generar firma para la actualización de firmware \(p. 1156\)](#)
- [Revise los siguientes pasos \(p. 1157\)](#)

Generar el archivo de actualización de firmware

El software LoRa Basics Station que se ejecuta en la puerta de enlace es capaz de recibir actualizaciones de firmware en la respuesta CUPS. Si no tiene ningún script proporcionado por el fabricante, consulte el siguiente script de actualización de firmware que se escribe para RakWireless Gateway basado en Raspberry Pi. Tenemos un script base y el nuevo binario de la estación, el archivo de versión ystation.confestán unidos a él.

Note

El script es específico de RAKWireless Gateway, por lo que tendrás que adaptarlo a tu aplicación en función de la puerta de enlace que estés utilizando.

Script base

A continuación se muestra un script base de ejemplo para RakWireless Gateway basado en Raspberry Pi. Puede guardar los siguientes comandos en un archivo `base.sh`, a continuación, ejecute el script en el terminal del navegador web de 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_version
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
```

Adición de scripts de carga

Al script base, añadimos el binario Basics Station, el version.txt que identifica la versión a la que se va a actualizar, y station.conf en un guión llamado addpayload.sh. Luego, ejecuta este guión.

```
*#!/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
```

Después de ejecutar estos scripts, puede ejecutar el siguiente comando en el terminal para generar el archivo de actualización de firmware, fwstation.

```
$ ./addpayload.sh station version.txt station.conf
```

Generar firma para la actualización de firmware

El software LoRa Basics Station proporciona actualizaciones de firmware firmadas con firmas ECDSA. Para admitir actualizaciones firmadas, necesitará lo siguiente:

- Firma que debe generarse mediante una clave privada ECDSA y menos de 128 bytes.
- La clave privada que se utiliza para la firma y debe almacenarse en la puerta de enlace con el nombre de archivo del formato sig-%d.key. Recomendamos utilizar el nombre del archivo sig-0.key.
- CRC de 32 bits a través de la clave privada.

La firma y el CRC se pasarán a la AWS IoT Core for LoRaWAN API. Para generar los archivos anteriores, puede utilizar el siguiente script gen.sh que se inspira en el estación básica ejemplo del repositorio de 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
```

La clave privada generada por el script debe guardarse en la puerta de enlace. El archivo de claves está en formato binario.

```
./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/p2ssgXIPNC0gZr+NzeTLpX+WfBo5tYWh5pQWN3AiBROen+XlIdMScv
AsfVfU/ZScJCalVNZh4esyS8mNIgA==

$ ls sig-0.key
sig-0.key

$ scp sig-0.key pi@192.168.1.11:/home/pi/Documents/basicstation/examples/iotwireless
```

Revise los siguientes pasos

Ahora que ha generado el firmware y la firma, vaya al siguiente tema para cargar el archivo de firmware, `fwstation`, en un bucket de Amazon S3. El depósito es un contenedor que almacenará el archivo de actualización de firmware como objeto. Puede agregar un rol de IAM que dará permiso al servidor CUPS para leer el archivo de actualización de firmware en el bucket de S3.

Cargue el archivo de firmware en un bucket de S3 y agregue una función de IAM

Puede utilizar Amazon S3 para crear un balde, que es un contenedor que puede almacenar el archivo de actualización de firmware. Puede cargar el archivo en el depósito de S3 y agregar un rol de IAM que permita al servidor CUPS leer el archivo de actualización desde el bucket. Para obtener más información acerca de Amazon S3, consulte [Introducción a Amazon S3](#).

El archivo de actualización de firmware que desea cargar depende de la puerta de enlace que utilice. Si ha seguido un procedimiento similar al descrito en[Generar el archivo de actualización de firmware y la firma \(p. 1154\)](#), subirá lafwstationarchivo generado mediante la ejecución de los scripts.

Para completar este procedimiento se necesitan aproximadamente 20 minutos.

Para cargar el archivo de firmware:

- [Crear un bucket de Amazon S3 y cargar el archivo de actualización \(p. 1158\)](#)
- [Crear un rol de IAM con permisos para leer el bucket de S3 \(p. 1158\)](#)
- [Revise los siguientes pasos \(p. 1160\)](#)

Crear un bucket de Amazon S3 y cargar el archivo de actualización

Creará un bucket de Amazon S3 mediante laAWS Management Consoley, a continuación, cargue el archivo de actualización de firmware en el depósito.

Creación de un bucket de S3

Para crear un bucket de S3, abra la[Consola de Amazon S3](#). Inicie sesión si aún no lo ha hecho y luego lleve a cabo todos los pasos que figuran a continuación.

1. Elija Create bucket (Crear bucket).
2. Escriba un nombre único y significativo para laNombre del bucket, (por ejemplo,iotwirelessfwupdate). Para obtener una convención de nomenclatura recomendada para su depósito, consulte<https://docs.aws.amazon.com/AmazonS3/latest/userguide/bucketnamingrules.html>.
3. Asegúrate de haber seleccionado elRegión de AWSseleccionado como el que utilizó para crear su puerta de enlace y dispositivo LoraWAN, y elBlock all public access (Bloquear todo el acceso público)se selecciona para que el depósito utilice los permisos predeterminados.
4. ElegirHabilitarparaControl de versiones del bucketque le ayudará a conservar varias versiones del archivo de actualización de firmware en el mismo bucket.
5. ConfirmarCifrado en el servidotoma el valorDeshabilitary eligeCreación de un bucket.

Cargue el archivo de actualización de firmware

Ahora puede ver el bucket en la lista de Buckets que se muestra enAWS Management Console. Elige tu depósito y completa los siguientes pasos para cargar el archivo.

1. Elija su bucket y luego elijaCargar.
2. ElegirAdición de archivoy, a continuación, cargue el archivo de actualización de firmware. Si ha seguido el procedimiento que se describe[Generar el archivo de actualización de firmware y la firma \(p. 1154\)](#), subirá lafwstation; de lo contrario, cargue el archivo proporcionado por el fabricante de la puerta de enlace.
3. Asegúrese de que todos los ajustes estén configurados como predeterminados. Asegúrese de queACL predefinidas detoma el valorprivatey eligeCargarpara cargar el archivo.
4. Copie el URI de S3 del archivo que ha subido. Elige tu depósito y verás que el archivo que has subido se muestra en la lista deObjetos. Elija el archivo y luego elijaCopiar URI S3. El URI tendrá un aspecto similar a:s3://iotwirelessfwupdate/fwstationsi ha nombrado su depósito de forma similar al ejemplo descrito anteriormente (fwstation). Usará el URI de S3 al crear la función de IAM.

Crear un rol de IAM con permisos para leer el bucket de S3

Ahora creará un rol y una política de IAM que darán permiso a CUPS para leer el archivo de actualización de firmware desde el bucket de S3.

Crear una política de IAM para su rol

Para crear una política de IAM para su AWS IoT Core for LoRaWAN rol de destino, abra el [Centro de políticas de la consola de IAM](#) y luego complete los pasos que se describen a continuación:

1. Elija Crear política, y elija la JSON Pestaña.
2. Elimine cualquier contenido del editor y pegue este documento de política. La política proporciona permisos para acceder a `iotwirelessbucket` y el archivo de actualización de firmware, `fwstation`, almacenado dentro de un 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. Elija Política de revisión, y en Nombre, introduzca un nombre para esta política (por ejemplo, `IoTWirelessFwUpdatePolicy`). Necesitará este nombre para utilizar en el siguiente procedimiento.
4. Elija Create Policy (Crear política).

Crear un rol de IAM con la política adjunta

Ahora creará un rol de IAM y asociará la política creada anteriormente para acceder al bucket de S3. Abra el icono [Centro de roles de la consola de IAM](#) y complete los pasos que se describen a continuación:

1. Elija Create role (Crear rol).
2. En Seleccione el tipo de entidad de confianza, elige Otra cuenta de AWS.
3. En ID de cuenta, introduzca su Cuenta de AWS ID y, a continuación, elija Siguiente: Permisos.
4. En el cuadro de búsqueda, escriba el nombre de la política de IAM que creó en el procedimiento anterior. Consulte la política de IAM (por ejemplo, `IoTWirelessFwUpdatePolicy`) que creaste anteriormente en los resultados de búsqueda y lo eliges.
5. Selecione Next (Siguiente): Etiquetas, y, a continuación, seleccione Siguiente: Consulte.
6. En Nombre del rol, introduzca el nombre de este rol (por ejemplo, `IoTWirelessFwUpdateRole`) y, a continuación, seleccione Creación de un rol.

Editar la relación de confianza del rol de IAM

En el mensaje de confirmación que aparece después de ejecutar el paso anterior, elija el nombre del rol que creó para editarlo. Editará la función para agregar la siguiente relación de confianza.

1. En la Resumen sección del rol que ha creado, elija la Relaciones de confianza Pestaña y, a continuación, seleccione Modificar relación de confianza.
2. En Policy Document, cambie la Principal propiedad para tener el aspecto del siguiente ejemplo.

```
"Principal": {  
    "Service": "iotwireless.amazonaws.com"  
},
```

Después de cambiar el Principal, el documento de política completo debe tener el aspecto del siguiente ejemplo.

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

3. Para guardar los cambios y salir, elija Actualizar política de confianza.
4. Obtenga el ARN de su rol. Elige tu rol de IAM y, en la sección Resumen, verás un ARN de rol, como, por ejemplo, arn:aws:iam::123456789012:role/IoTWirelessFwUpdateRole. Copie esto ARN de rol.

Revise los siguientes pasos

Ahora que ha creado el bucket de S3 y un rol de IAM que permite al servidor CUPS leer el bucket de S3, vaya al tema siguiente para programar y ejecutar la actualización del firmware. Mantenga la S3 URLyARN de rolque ha copiado anteriormente para poder introducirlos para crear una definición de tarea que se ejecutará para llevar a cabo la actualización del firmware.

Programar y ejecutar la actualización del firmware mediante una definición de tarea

Puede utilizar una definición de tarea para incluir detalles sobre la actualización del firmware y definir la actualización. AWS IoT Core for LoRaWAN proporciona una actualización de firmware basada en la información de los tres campos siguientes asociados a la puerta de enlace.

- estación

La versión y el tiempo de compilación del software Basics Station. Para identificar esta información, también puede generarla utilizando el software Basics Station que ejecuta la puerta de enlace (por ejemplo, 2.0.5(rpi/std) 2021-03-09 03:45:09).

- Versión de paquete

La versión del firmware, especificada por el archivo `version.txt` en la puerta de enlace. Si bien es posible que esta información no esté presente en la puerta de enlace, la recomendamos como forma de definir la versión de firmware (por ejemplo, 1.0.0).

- Modelo

Plataforma o modelo que utiliza la puerta de enlace (por ejemplo, Linux).

Para completar este procedimiento se necesitan 20 minutos.

Para completar este procedimiento:

- [Obtenga la versión actual en ejecución en su puerta de enlace \(p. 1161\)](#)
- [Creación de una definición de tarea de gateway inalámbrica \(p. 1162\)](#)
- [Ejecute la tarea de actualización de firmware y haga un seguimiento del progreso \(p. 1163\)](#)

Obtenga la versión actual en ejecución en su puerta de enlace

Para determinar la elegibilidad de la puerta de enlace para una actualización de firmware, el servidor CUPS comprueba los tres campos, `Station`, `PackageVersion`, y `Model`, para una coincidencia cuando la puerta de enlace los presenta durante una solicitud CUPS. Cuando utiliza una definición de tarea, estos campos se almacenan como parte del `CurrentVersion`.

Puede utilizar el [AWS IoT Core for LoRaWAN API](#) o [AWS CLI](#) para obtener el `CurrentVersion` para su gateway (gateway). Los siguientes comandos muestran cómo obtener esta información mediante la CLI.

1. Si ya ha aprovisionado una puerta de enlace, puede obtener información sobre la puerta de enlace mediante el `get-wireless-gateway` comando.

```
aws iotwireless get-wireless-gateway \
    --identifier 5a11b0a85a11b0a8 \
    --identifier-type GatewayEui
```

A continuación se muestra una salida de ejemplo del 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. Uso del ID del gateway inalámbrico que ha informado `get-wireless-gateway`, puede usar la [obtener información de firmware de puerta de enlace inalámbrica](#) comando para obtener el `CurrentVersion`.

```
aws iotwireless get-wireless-gateway-firmware-information \
    --id "3039b406-5cc9-4307-925b-9948c63da25b"
```

A continuación se muestra un ejemplo de salida para el comando, con información de los tres campos que muestra el `CurrentVersion`.

```
{
    "LoRaWAN": {
        "CurrentVersion": {
            "PackageVersion": "1.0.0",
            "Model": "rpi",
            "Station": "2.0.5(rpi/std) 2021-03-09 03:45:09"
        }
    }
}
```

Creación de una definición de tarea de gateway inalámbrica

Al crear la definición de tarea, le recomendamos que especifique la creación automática de tareas mediante el parámetro `AutoCreateTasks`. `AutoCreateTasks` se aplica a cualquier puerta de enlace que coincida con los tres parámetros mencionados anteriormente. Si este parámetro está deshabilitado, los parámetros deben asignarse manualmente a la puerta de enlace.

Puede crear la definición de tareas de puerta de enlace inalámbrica mediante la AWS IoT Core for LoRaWAN API o AWS CLI. Los siguientes comandos muestran cómo crear la definición de tarea mediante la CLI.

1. Cree un archivo `input.json`, que contendrá la información que se va a pasar a la `CreateWirelessGatewayTaskDefinitionAPI`. En el navegador `input.json`, proporcione la siguiente información que ha obtenido anteriormente:

- `UpdateDataSource`

Proporcione el enlace al objeto que contiene el archivo de actualización de firmware que ha cargado en el bucket de S3. (por ejemplo, `s3://iotwirelessfwupdate/fwstation`).

- Función de actualización de datos

Proporcione el enlace al ARN de rol para el rol de IAM que ha creado, que proporciona permisos para leer el bucket de S3. (por ejemplo, `arn:aws:iam::123456789012:role/IoTWirelessFwUpdateRole`).

- `Sigkey CRC` y `UpdateSignature`

Es posible que el fabricante de la puerta de enlace proporcione esta información, pero si ha seguido el procedimiento descrito en [Generar el archivo de actualización de firmware y la firma \(p. 1154\)](#), encontrará esta información al generar la firma.

- Versión actual

Proporcione la `CurrentVersion` obtenida anteriormente ejecutando el `get-wireless-gateway-firmware-information` comando.

```
cat input.json
```

A continuación se muestra el contenido del `input.json`.

```
{  
    "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. Pasar el archivo `input.json` al comando [create-wireless-gateway-task-definition](#) para crear la definición de tarea.

```
aws iotwireless create-wireless-gateway-task-definition \
--cli-input-json file://input.json
```

A continuación se muestra la salida del comando.

```
{
  "Id": "4ac46ff4-efc5-44fd-9def-e8517077bb12",
  "Arn": "arn:aws:iotwireless:us-
east-1:231894231068:WirelessGatewayTaskDefinition/4ac46ff4-efc5-44fd-9def-e8517077bb12"
}
```

Ejecute la tarea de actualización de firmware y haga un seguimiento del progreso

La puerta de enlace está lista para recibir la actualización del firmware y, una vez encendida, se conecta al servidor CUPS. Cuando el servidor CUPS encuentra una coincidencia en la versión de la puerta de enlace, programa una actualización del firmware.

Una tarea es una definición de tarea en proceso. Tal como especificó la creación automática de tareas mediante la configuración `AutoCreateTasks` a `True`, la tarea de actualización de firmware se inicia en cuanto se encuentra una puerta de enlace coincidente.

Puede realizar un seguimiento del progreso de la tarea mediante la `GetWirelessGatewayTaskAPI`. Cuando ejecutas el comando [get-wireless-gateway-task](#) la primera vez, mostrará el estado de la tarea como `IN_PROGRESS`.

```
aws iotwireless get-wireless-gateway-task \
--id 1352172b-0602-4b40-896f-54da9ed16b57
```

A continuación se muestra la salida del 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"
}
```

Cuando ejecute el comando la próxima vez, si la actualización del firmware entra en vigor, mostrará los campos actualizados, `Package`, `Version`, y `Model` y el estado de la tarea cambia a `COMPLETED`.

```
aws iotwireless get-wireless-gateway-task \
--id 1352172b-0602-4b40-896f-54da9ed16b57
```

A continuación se muestra la salida del 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"
}
```

En este ejemplo, le mostramos la actualización del firmware utilizando la puerta de enlace RakWireless basada en Raspberry Pi. El script de actualización de firmware detiene la BasicStation en ejecución para almacenar la actualizaciónPackage, Version, yModelcampos para que BasicStation tenga que reiniciarse.

```
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
```

Si se produce un error en la actualización del firmware, verá el estado deFIRST_RETRYdesde el servidor CUPS y la puerta de enlace envía la misma solicitud. Si el servidor CUPS no puede conectarse a la gateway después de unSECOND_RETRY, mostrará el estado deFAILED.

Después de la tarea anteriorCOMPLETEDFAILED, elimine la antigua tarea mediante el[delete-wireless-gateway-task](#)antes de iniciar uno nuevo.

```
aws iotwireless delete-wireless-gateway-task \
--id 1352172b-0602-4b40-896f-54da9ed16b57
```

Administración de dispositivos con AWS IoT Core for LoRaWAN

LoRaLos dispositivos WAN se comunicanAWS IoT Core for LoRaWANatravés deLoRapuertas de enlace WAN. Adición de dispositivos aAWS IoT Core for LoRaWANDoneAWS IoTprocesar los mensajes recibidos de los dispositivos para que los utilicenAWS IoTy otros servicios de.

A continuación se presentan algunas consideraciones importantes al utilizar los dispositivos conAWS IoT Core for LoRaWAN. Para obtener información acerca de cómo agregar su dispositivo aAWS IoT Core for LoRaWAN, consulte[Incorporación de sus dispositivos aAWS IoT Core for LoRaWAN \(p. 1129\)](#).

Consideraciones sobre el dispositivo

Al seleccionar un dispositivo con el que desea usar para comunicarse conAWS IoT Core for LoRaWAN, considere lo siguiente.

- Sensores disponibles
- Capacidad de la batería
- Consumo de energía
- Costo
- Tipo de antena y rango de transmisión

Uso de dispositivos con puertas de enlace calificadas paraAWS IoT Core for LoRaWAN

Los dispositivos que utiliza se pueden emparejar con puertas de enlace inalámbricas calificadas para su uso conAWS IoT Core for LoRaWAN. Puede encontrar estas puertas de enlace y kits de desarrollador en

elAWS Catálogo de dispositivos de socios. También recomendamos que considere la proximidad de estos dispositivos a sus puertas de enlace. Para obtener más información, consulte [Uso de puertas de enlace calificadas desde elAWS Catálogo de dispositivos asociados \(p. 1151\)](#).

LoRaVersión WAN

AWS IoT Core for LoRaWANes compatible con todos los dispositivos que cumplen los requisitos 1.0.x o 1.1LoRaEspecificaciones WAN estandarizadas porLoRaAlianza.

Modos de activación

Antes de tuLoRaEl dispositivo WAN puede enviar datos de enlace ascendente, debe completar un proceso llamadoactivaciónounirseprocedimiento. Para activar el dispositivo, puede utilizar OTAA (activación por aire) o ABP (activación por personalización). Le recomendamos que utilice OTAA para activar el dispositivo porque se generan nuevas claves de sesión para cada activación, lo que lo hace más seguro.

La especificación de su dispositivo inalámbrico se basa en laLoRaVersión WAN y modo de activación, que determinan las claves raíz y las claves de sesión generadas para cada activación. Para obtener más información, consulte [Añada la especificación de su dispositivo inalámbrico aAWS IoT Core for LoRaWANUs de la consola \(p. 1130\)](#).

Clases de dispositivo

LoRaLos dispositivos WAN pueden enviar mensajes de enlace ascendente en cualquier momento. Escuchar mensajes de enlace descendente consume capacidad de la batería y reduce la duración de la batería. LaLoRaEl protocolo WAN especifica tres clases deLoRaDispositivos WAN.

- Los dispositivos de clase A duermen la mayor parte del tiempo y escuchan mensajes de enlace descendente solo durante un breve período de tiempo. Estos dispositivos son en su mayoría sensores alimentados por batería con una duración de la batería de hasta 10 años.
- Los dispositivos de clase B pueden recibir mensajes en ranuras de enlace descendente programadas. Estos dispositivos son en su mayoría actuadores alimentados por batería.
- Los dispositivos de clase C nunca se duermen y escuchan continuamente los mensajes entrantes, por lo que no hay mucho retraso en la recepción de los mensajes. Estos dispositivos son en su mayoría actuadores alimentados por la red.

Para obtener más información sobre estas consideraciones de dispositivos inalámbricos, consulte los recursos mencionados en[Más información sobre LoRaWAN \(p. 1118\)](#).

Gestiona la comunicación entre tuLoRaDispositivos WAN yAWS IoT

Una vez que hayas conectado tuLoRaDispositivo WAN aAWS IoT Core for LoRaWAN, los dispositivos pueden empezar a enviar mensajes a la nube. Los mensajes de enlace ascendente son mensajes que se envían desde su dispositivo y se reciben porAWS IoT Core for LoRaWAN. SusLoRaLos dispositivos WAN pueden enviar mensajes de enlace ascendente en cualquier momento, que luego se reenvían a otrosServicio de AWSs y aplicaciones alojadas en la nube. Mensajes enviados desdeAWS IoT Core for LoRaWANy otrosServicio de AWSs y aplicaciones para sus dispositivos se denominan mensajes de enlace descendente.

A continuación se muestra cómo puede ver y administrar los mensajes de enlace ascendente y descendente que se envían entre sus dispositivos y la nube. Puede mantener una cola de mensajes de enlace descendente y enviarlos a sus dispositivos en el orden en que se agregaron a la cola.

Temas

- [Ver formato de los mensajes de enlace ascendente enviados desdeLoRaDispositivos WAN \(p. 1166\)](#)
- [Poner en cola los mensajes de enlace descendente para enviarlosLoRaDispositivos WAN \(p. 1168\)](#)

Ver formato de los mensajes de enlace ascendente enviados desdeLoRaDispositivos WAN

Una vez que hayas conectado tuLoRaDispositivo WAN aAWS IoT Core for LoRaWAN, puedes observar el formato del mensaje de enlace ascendente que recibirás desde tu dispositivo inalámbrico.

Antes de poder observar los mensajes de enlace ascendente

Debe haber incorporado su dispositivo inalámbrico y conectado el dispositivo aAWS IoT para que pueda transmitir y recibir datos. Para obtener información acerca de la incorporación de su dispositivo enAWS IoT Core for LoRaWAN, consulte [Incorporación de sus dispositivos aAWS IoT Core for LoRaWAN \(p. 1129\)](#).

¿Qué contienen los mensajes de enlace ascendente?

LoRaSe conectan dispositivos WAN aAWS IoT Core for LoRaWANmedianteLoRapuertas de enlace WAN. El mensaje de enlace ascendente que recibe del dispositivo contendrá la siguiente información.

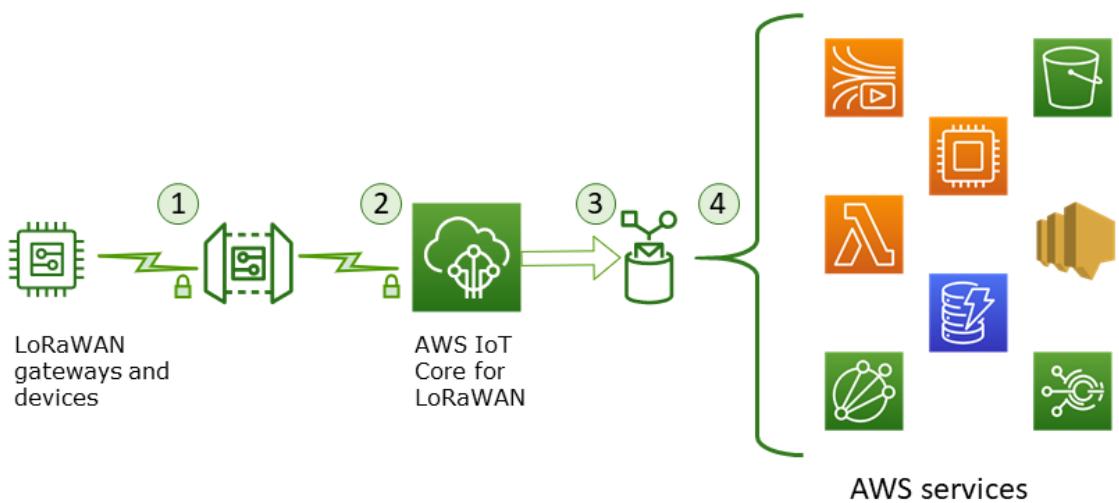
- Datos de carga útil que corresponden al mensaje de carga útil cifrada que se envía desde el dispositivo inalámbrico.
- Metadatos inalámbricos que incluyen:
 - Información de dispositivo, comoDevEui, la velocidad de datos y el canal de frecuencia en el que funciona el dispositivo.
 - Parámetros adicionales opcionales e información de puerta de enlace para las puertas de enlace conectadas al dispositivo. Los parámetros de puerta de enlace incluyen la EUI de la puerta de enlace, el SNR y la RSSI.

Mediante los metadatos inalámbricos, puede obtener información útil sobre el dispositivo inalámbrico y los datos que se transmiten entre su dispositivo yAWS IoT. Por ejemplo, puede utilizar laAckedMessageIdpara comprobar si el dispositivo ha recibido el último mensaje de enlace descendente confirmado. Opcionalmente, si decides incluir la información de la puerta de enlace, puedes identificar si quieras cambiar a un canal de puerta de enlace más sólido que esté más cerca de tu dispositivo.

¿Cómo observar los mensajes de enlace ascendente?

Una vez que hayas incorporado el dispositivo, puedes usar el[Cliente de prueba MQTT](#)en el[Pruebas](#)Página de laAWS IoTpara suscribirse al tema que especificó al crear su destino. Empezarás a ver los mensajes después de que el dispositivo esté conectado y comience a enviar datos de carga útil.

Este diagrama identifica los elementos clave de unLoRaSistema WAN conectado aAWS IoT Core for LoRaWAN, que muestra el plano de datos principal y cómo fluyen los datos a través del sistema.



Cuando el dispositivo inalámbrico empieza a enviar datos de enlace ascendente,AWS IoT Core for LoRaWANenvuelve la información de metadatos inalámbricos con la carga útil y, a continuación, la envía a suAWS aplicaciones.

Ejemplo de mensaje de enlace superior

En el ejemplo siguiente se muestra el formato del mensaje de enlace superior recibido de su dispositivo.

```
{  
    "WirelessDeviceId": "5b58245e-146c-4c30-9703-0ca942e3ff35",  
    "PayloadData": "Cc48AAAAAAAAAA=",  
    "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"  
        }  
    }  
}
```

```
}
```

En la tabla siguiente se muestra una descripción de los campos utilizados en los metadatos del vínculo superior:

LoRaCampos de mensaje de enlace ascendente de

Parámetro	Descripción	Tipo	Obligatorio
WirelessDeviceID	ID del dispositivo inalámbrico que envía los datos.	Cadena	Sí
PayloadData	Mensaje binario recibido del dispositivo, codificado en base64.	Cadena	Sí
WirelessMetadata	Metadatos sobre elLoRaDispositivo WAN y la solicitud de mensaje. Esto incluye información como los identificadores de dispositivo, la velocidad de datos y código, la marca de hora del mensaje, si ADR (velocidad de datos adaptativa) está habilitado y los metadatos de la puerta de enlace.	Enumeración	No

Excluir metadatos de puerta de enlace de metadatos de vínculos ascendentes

Si desea excluir la información de metadatos de la puerta de enlace de los metadatos de enlace superior, deshabilite laAddGwMetadatoscuando se crea el perfil de servicio. Para obtener información acerca de la desactivación de este parámetro, consulte[Añada perfiles de servicio \(p. 1133\)](#).

En este caso, no verá laGatewaysde los metadatos del vínculo superior, como se ilustra en el siguiente ejemplo.

```
{
    "WirelessDeviceId": "0d9a439b-e77a-4573-a791-49d5c0f4db95",
    "PayloadData": "AAAAAAA//8=",
    "WirelessMetadata": {
        "LoRaWAN": {
            "ClassB": false,
            "CodeRate": "4/5",
            "DataRate": "1",
            "DevAddr": "01920f27",
            "DevEui": "ffffffff10000163b0",
            "FCnt": 1,
            "FFPort": 5,
            "Timestamp": "2021-04-29T05:19:43.646Z"
        }
    }
}
```

Poner en cola los mensajes de enlace descendente para enviarlosLoRaDispositivos WAN

Aplicaciones alojadas en la nube y otrosServicio de AWSs puede enviar mensajes de enlace descendente a sus dispositivos inalámbricos. Los mensajes de enlace descendente son mensajes que se envían desdeAWS IoT Core for LoRaWANa su dispositivo inalámbrico. Puedes programar y enviar mensajes de enlace descendente para cada dispositivo en el que hayas incorporadoAWS IoT Core for LoRaWAN.

Si tiene varios dispositivos para los que desea enviar un mensaje de enlace descendente, puede utilizar un grupo de multidifusión. Los dispositivos de un grupo de multidifusión comparten la misma dirección de multidifusión, que luego se distribuye a todo un grupo de dispositivos de destinatarios. Para obtener más información, consulte [Crear grupos de multidifusión para enviar una carga útil de enlace descendente a varios dispositivos \(p. 1172\)](#).

Cómo funciona una cola de mensajes de enlace descendente

La clase de dispositivo de tuLoRaEl dispositivo WAN determina cómo se envían los mensajes de la cola al dispositivo. Los dispositivos de clase A envían un mensaje de enlace ascendente aAWS IoT Core for LoRaWANpara indicar que el dispositivo está disponible para recibir mensajes de enlace descendente. Los dispositivos de clase B pueden recibir mensajes en ranuras de enlace descendente normales. Los dispositivos de clase C pueden recibir mensajes de enlace descendente en cualquier momento. Para obtener más información acerca de las clases de dispositivos, consulte[Clases de dispositivo \(p. 1165\)](#).

A continuación se muestra cómo se ponen en cola los mensajes y se envían a los dispositivos de clase A.

1. AWS IoT Core for LoRaWANalmacena en búfer el mensaje de enlace descendente que ha agregado a la cola con el puerto de trama, los datos de carga útil y los parámetros del modo de confirmación especificados mediante elAWS IoTConsola de oAWS IoT WirelessAPI.
2. SusLoRaEl dispositivo WAN envía un mensaje de enlace ascendente para indicar que está en línea y puede comenzar a recibir mensajes de enlace descendente.
3. Si ha añadido más de un mensaje de enlace descendente a la cola,AWS IoT Core for LoRaWANenvía el primer mensaje de enlace descendente de la cola al dispositivo con la marca de confirmación (ACK) establecida.
4. El dispositivo envía un mensaje de enlace ascendente aAWS IoT Core for LoRaWANinmediatamente, o se duerme hasta el siguiente mensaje de enlace ascendente e incluye la marca ACK en el mensaje.
5. CuandoAWS IoT Core for LoRaWANrecibe el mensaje de enlace ascendente con el indicador ACK, borra el mensaje de enlace descendente de la cola, lo que indica que el dispositivo ha recibido correctamente el mensaje de enlace descendente. Si falta el indicador ACK en el mensaje de enlace ascendente después de verificarlo tres veces, el mensaje se descarta.

Realice operaciones de cola de enlaces descendentes mediante la consola

Puede utilizar elAWS Management Consolepara poner en cola mensajes de enlace descendente y borrar mensajes individuales, o toda la cola, según sea necesario. En el caso de los dispositivos de clase A, después de recibir un enlace ascendente del dispositivo para indicar que está conectado, los mensajes en cola se envían al dispositivo. Después de enviar el mensaje, se borra automáticamente de la cola.

Cola de mensajes de enlace descendente

Para crear una cola de mensajes de enlace descendente

1. Vaya a la[Hub de dispositivos delAWS IoTconsola](#)y elige el dispositivo para el que quieras poner en cola los mensajes de enlace descendente.
2. En el navegadorMensaje de enlace de bajadade la página de detalles del dispositivo, elijaCola de mensajes de enlace descendente.
3. Especifique los siguientes parámetros para configurar el mensaje de enlace descendente:
 - Fport: Elija el puerto de trama para que el dispositivo se comuniqueAWS IoT Core for LoRaWAN.
 - Carga: Especifique el mensaje de carga útil que desea enviar a su dispositivo. El tamaño de carga máximo es 242 bytes. Si la velocidad de datos adaptativa (ADR) está habilitada,AWS IoT Core for LoRaWANlo utiliza para elegir la velocidad de datos óptima para el tamaño de su carga útil. Puede optimizar aún más la velocidad de datos según sea necesario.
 - Modo Confirmación: Confirma si tu dispositivo ha recibido el mensaje de enlace descendente. Si un mensaje requiere este modo, verás un mensaje de enlace ascendente con el indicador ACK en el flujo de datos y el mensaje se borrará de la cola.

4. Para añadir el mensaje de enlace descendente a la cola, elijaEnviar.

El mensaje de enlace descendente se ha añadido a la cola. Si no ve el mensaje o recibe un error, puede solucionarlo tal y como se describe en[Solución de problemas de errores de cola de mensajes de enlace descendente \(p. 1171\)](#).

Note

Después de agregar el mensaje de enlace descendente a la cola, ya no podrá editar los parámetrosFport,Carga, yModo Confirmación. Para enviar un mensaje de enlace descendente con valores diferentes para estos parámetros, puede eliminar este mensaje y poner en cola un nuevo mensaje de enlace descendente con los valores de los parámetros actualizados.

La cola enumera los mensajes de enlace descendente que ha agregado. Para ver la carga útil de los mensajes de enlace ascendente y descendente que se intercambian entre sus dispositivos yAWS IoT Core for LoRaWAN, puede utilizar el analizador de red. Para obtener más información, consulte[Supervisión de su flota de recursos inalámbricos en tiempo real mediante el analizador de redes \(p. 1193\)](#).

Lista de colas de mensajes de enlace descendente

El mensaje de enlace descendente que creó se añade a la cola. Cada mensaje de enlace descendente posterior se añade a la cola después de este mensaje. Puede ver una lista de mensajes de enlace descendente en laMensaje de enlace de bajadade la página de detalles del dispositivo. Después de recibir un enlace ascendente, los mensajes se envían al dispositivo. Una vez que el dispositivo haya recibido un mensaje de enlace descendente, se eliminará de la cola. El siguiente mensaje se mueve hacia arriba en la cola para enviarlo al dispositivo.

Eliminar mensajes de enlace descendente individuales o borrar toda la cola

Cada mensaje de enlace descendente se borra de la cola automáticamente después de enviarlo al dispositivo. También puede eliminar mensajes individuales o borrar toda la cola de vínculos descendentes. Estas acciones no se pueden deshacer.

- Si encuentras mensajes en la cola que no quieras enviar, elige los mensajes y eligeBorrar.
- Si no quieres enviar mensajes de la cola a tu dispositivo, puedes borrar toda la cola seleccionandoBorrar cola de vínculos descendentes.

Realizar operaciones de cola de vínculos descendentes mediante la API

Puede utilizar elAWS IoT WirelessAPI para poner en cola mensajes de enlace descendente y borrar mensajes individuales, o toda la cola, según sea necesario.

Cola de mensajes de enlace descendente

Para crear una cola de mensajes de enlace descendente, utilice la[SendDataToWirelessDevice](#)Operación de la API de[send-data-to-wireless-device](#)Command de la 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}
```

El resultado de ejecutar este comando genera unMessageIdpara el mensaje de enlace descendente. En algunos casos, incluso si recibe laMessageId, los paquetes se pueden dejar caer. Para obtener

más información acerca de cómo resolver el error, consulte[Solución de problemas de cola de mensajes de enlace descendente \(p. 1171\)](#).

```
{  
    MessageId: "6011dd36-0043d6eb-0072-0008"  
}
```

Lista de mensajes de enlace descendente en la cola

Para listar todos los mensajes de enlace descendente de la cola, utilice [laListQueuedMessagesOperación de la API de list-queued-messagesCommand de la CLI](#).

```
aws iotwireless list-queued-messages
```

De forma predeterminada, se muestran un máximo de 10 mensajes de enlace descendente al ejecutar este comando.

Eliminar mensajes de enlace descendente individuales o borrar toda la cola

Para eliminar mensajes individuales de la cola o borrar toda la cola, utilice [laDeleteQueuedMessagesOperación de la API de delete-queued-messagesCommand de la CLI](#).

- Para eliminar mensajes individuales, proporcione la `messageID` para los mensajes que desea eliminar de su dispositivo inalámbrico, especificados por `elwirelessDeviceId`.
- Para borrar toda la cola de vínculos descendentes, especifique `messageID` como `*` para su dispositivo inalámbrico, especificado por `elwirelessDeviceId`.

Solución de problemas de errores de cola de mensajes de enlace descendente

Estas son algunas cosas que debes comprobar si no ves los resultados esperados:

- Los mensajes de enlace descendente no aparecen en la AWS IoT consola

Si no ves el mensaje de enlace descendente en la cola después de agregarlo como se describe en[Realice operaciones de cola de enlaces descendentes mediante la consola \(p. 1169\)](#), puede deberse a que el dispositivo no ha completado un proceso llamado activación o procedimiento de unión. Este procedimiento se completa cuando el dispositivo se incorpora con AWS IoT Core for LoRaWAN. Para obtener más información, consulte [Añada la especificación de su dispositivo inalámbrico a AWS IoT Core for LoRaWANUsos de la consola \(p. 1130\)](#).

Después de incorporar el dispositivo a AWS IoT Core for LoRaWAN, puede supervisar su dispositivo para comprobar si la unión y la reunión se han realizado correctamente mediante el analizador de red o AmazonCloudWatch. Para obtener más información, consulte [Monitoreo y registro de AWS IoT Wirelessusos de Amazon CloudWatch \(p. 1220\)](#).

- Falta paquetes de mensajes de enlace descendente al utilizar la API

Cuando utiliza `elSendDataToWirelessDevice` Operación de la API, la API devuelve un único `MessageId`. Sin embargo, no puede confirmar si su LoRaEl dispositivo WAN ha recibido el mensaje de enlace descendente. Los paquetes de enlace descendente pueden descartarse en casos como cuando el dispositivo no ha completado el procedimiento de unión. Para obtener más información acerca de cómo resolver este error, consulte la sección anterior.

- Error ARN faltante al enviar un mensaje de enlace descendente

Al enviar un mensaje de enlace descendente a su dispositivo desde la cola, puede recibir un error de nombre de recurso de Amazon (ARN) que falta. Este error puede producirse porque el destino no se

ha especificado correctamente para el dispositivo que recibe el mensaje de enlace descendente. Para solucionar este error, consulte los detalles de destino de su dispositivo.

Crear grupos de multidifusión para enviar una carga útil de enlace descendente a varios dispositivos

Para enviar una carga útil de enlace descendente a varios dispositivos, cree un grupo de multidifusión. Mediante la multidifusión, un origen puede enviar datos a una única dirección de multidifusión, que luego se distribuye a todo un grupo de dispositivos destinatarios.

Los dispositivos de un grupo de multidifusión comparten la misma dirección de multidifusión, claves de sesión y contador de fotogramas. Mediante el uso de las mismas claves de sesión, los dispositivos de un grupo de multidifusión pueden descifrar el mensaje cuando se inicia una transmisión de enlace descendente. Un grupo de multidifusión solo admite enlaces descendentes. No confirma si los dispositivos han recibido la carga útil del enlace descendente.

con AWS IoT Core for LoRaWAN los grupos de multidifusión, puedes:

- Filtra la lista de dispositivos mediante el perfil de dispositivo, RFRegion o clase de dispositivo y, a continuación, agrega estos dispositivos a un grupo de multidifusión.
- Programe y envíe uno o más mensajes de carga útil de vínculos descendentes a dispositivos de un grupo de multidifusión, en un plazo de distribución de 48 horas.
- Haga que los dispositivos cambien temporalmente al modo Clase B o clase C al inicio de la sesión de multidifusión para recibir el mensaje de enlace descendente.
- Supervise la configuración de su grupo de multidifusión y el estado de sus dispositivos, así como solucione cualquier problema.
- Utilice actualizaciones de firmware por aire (FUOTA) para implementar de forma segura actualizaciones de firmware en dispositivos de un grupo de multidifusión.

AWS IoT Core for LoRaWAN el soporte de FUOTA y grupos de multidifusión se basa en el [LoRaAlliance](#) especificaciones siguientes:

- LoRaEspecificación de configuración de multidifusión remota de WAN, TS005-2.0.0
- LoRaEspecificación de transporte de bloques de datos fragmentados WAN, TS004-2.0.0
- LoRaEspecificación de sincronización de reloj de capa de aplicaciones WAN, TS003-2.0.0

Note

AWS IoT Core for LoRaWAN realiza automáticamente la sincronización del reloj del dispositivo según el [LoRaEspecificación de alianza](#). Función de uso `AppTimeReq`, responde la hora del lado del servidor a los dispositivos que lo solicitan mediante `ClockSync` señalización.

A continuación se muestra cómo crear un grupo de multidifusión y programar un mensaje de enlace descendente.

Temas

- [Preparar dispositivos para la configuración de multidifusión y FUOTA \(p. 1173\)](#)
- [Cree grupos de multidifusión y agregue dispositivos al grupo \(p. 1175\)](#)
- [Supervisar y solucionar problemas del grupo de multidifusión y de los dispositivos del grupo \(p. 1179\)](#)
- [Programar un mensaje de enlace descendente para enviarlo a los dispositivos del grupo de multidifusión \(p. 1181\)](#)

Preparar dispositivos para la configuración de multidifusión y FUOTA

Cuando añades tu dispositivo inalámbrico a AWS IoT Core for LoRaWAN, puedes preparar su dispositivo inalámbrico para la configuración de multidifusión y la configuración de FUOTA mediante la consola o la CLI. Si va a realizar esta configuración por primera vez, se recomienda utilizar la consola. Para administrar el grupo de multidifusión y agregar o quitar varios dispositivos del grupo, recomendamos utilizar la CLI para administrar un gran número de recursos.

GenAppClave y Fports

Cuando agregue su dispositivo inalámbrico, antes de poder agregar los dispositivos a grupos de multidifusión o realizar actualizaciones de FUOTA, configure los siguientes parámetros. Antes de configurar estos parámetros, asegúrese de que sus dispositivos admiten FUOTA y multidifusión y que la especificación de su dispositivo inalámbrico sea OTAA v1.1 o TAAv1.0.x.

- **GenAppKey:** para dispositivos compatibles con el LoRaWAN versión 1.0.x y para utilizar grupos de multidifusión, el GenAppKey es la clave raíz específica del dispositivo de la que se derivan las claves de sesión del grupo de multidifusión.

Note

Para los dispositivos LoRaWAN que utilizan la especificación inalámbrica OTAA v1.1, el AppKey se utiliza con la misma finalidad que la GenAppKey.

Para configurar los parámetros para iniciar la transferencia de datos, AWS IoT Core for LoRaWAN distribuye las claves de sesión con los dispositivos finales. Para obtener más información acerca de las versiones LoRaWAN, consulte [LoRaVersión WAN \(p. 1165\)](#).

Note

AWS IoT Core for LoRaWAN almacena la GenAppKey en formato cifrado.

- **FPorts:** Según el LoRaWAN Especificaciones WAN para FUOTA y grupos multidifusión, AWS IoT Core for LoRaWAN asigna los valores predeterminados para los siguientes campos de FPort parámetro. Si ya ha asignado alguna de las siguientes opciones FPort valores y, a continuación, puede elegir otro valor disponible, de 1 a 223.
 - **Multicast:** 200

Este FPort value se utiliza para grupos de multidifusión.

- **FUOTA:** 201

Este FPort value se utiliza para FUOTA.

- **ClockSync:** 202

Este FPort valor se utiliza para la sincronización del reloj.

Perfiles de dispositivo para multidifusión y FUOTA

Al inicio de una sesión de multidifusión, se utiliza una ventana de distribución de clase B o clase C para enviar el mensaje de enlace descendente a los dispositivos del grupo. Los dispositivos que agregue para multidifusión y FUOTA deben admitir los modos de operación clase B o clase C. Según la clase de dispositivo que admite el dispositivo, elija un perfil de dispositivo para el dispositivo que tenga habilitado uno o ambos modos de clase B o clase C.

Para obtener información sobre los perfiles de dispositivos, consulte [Añadir perfiles a AWS IoT Core for LoRaWAN \(p. 1132\)](#).

Preparar dispositivos para multidifusión y FUOTA mediante la consola

Para especificar los fPorts y GenAppParámetros clave para la configuración de multidifusión y FUOTA mediante la consola:

1. Vaya a la [.Hub de dispositivos del AWS IoTconsolay](#) eligeAñadir dispositivo inalámbrico.
2. Elija el iconoEspecificación de dispositivo inalámbrico. El dispositivo debe utilizar OTAA para la activación del dispositivo. Cuando elige OTAA v1.0.x u OTAA v1.1, unConfiguración FUOTA: opcionalaparece la sección.
3. Introduzca los parámetros EUI (identificador único extendido) de su dispositivo inalámbrico.
4. Expanda laConfiguración FUOTA: opcionalsección y, a continuación, elijaEste dispositivo admite actualizaciones de firmware por aire (FUOTA). Ahora puede ingresar alFportvalores para multidifusión, FUOTA y sincronización de reloj. Si eligióTAA v1.0.xpara la especificación del dispositivo inalámbrico, introduzca laGenAppClave.
5. Añadir su dispositivo aAWS IoT Core for LoRaWANeligiendo sus perfiles y un destino para enrutar mensajes. Para el perfil de dispositivo vinculado al dispositivo, asegúrate de seleccionar uno o ambosSoporta clase BoSoporta clase Cmodos.

Note

Para especificar los parámetros de configuración de FUOTA, debe usar la[Hub de dispositivos del AWS IoTconsola](#). Estos parámetros no aparecen si incorporas tus dispositivos mediante elIntroPágina de laAWS IoTconsola de .

Para obtener más información acerca de la especificación del dispositivo inalámbrico y la incorporación de su dispositivo, consulte[Añade tu dispositivo inalámbrico aAWS IoT Core for LoRaWAN \(p. 1130\)](#).

Note

Puede especificar estos parámetros solo cuando crea el dispositivo inalámbrico. No puedes cambiar ni especificar parámetros al actualizar un dispositivo existente.

Preparar dispositivos para multidifusión y FUOTA mediante la operación API

Para utilizar grupos de multidifusión o para realizar actualizaciones de FUOTA, configure estos parámetros mediante la[createWirelessDeviceOperación de la API de](#)`create-wireless-device`Command de la CLI. Además de especificar la clave de aplicación y los parámetros de fPorts, asegúrese de que el perfil del dispositivo vinculado al dispositivo admite uno o ambos modos clase B o clase C.

Puede proporcionar un.jsonarchivo como entrada a la`create-wireless-device`comando.

```
aws iotwireless create-wireless-device \
--cli-input-json file://input.json
```

donde:

Contenido de 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 obtener información acerca de los comandos de la CLI que puede usar, consulte[AWS CLI Referencia de de](#).

Note

Después de especificar los valores de estos parámetros, no puede actualizarlos mediante la `UpdateWirelessDevice`Operación API. En su lugar, puede crear un nuevo dispositivo con los valores de los parámetros `GenAppKey`y `FPorts`.

Para obtener información sobre los valores especificados para estos parámetros, puede utilizar el `GetWirelessDevice`Operación de la API de `get-wireless-device`Command de la CLI.

Pasos siguientes

Después de configurar los parámetros, puede crear grupos de multidifusión y tareas de FUOTA para enviar carga útil de enlace descendente o actualizar el firmware de su LoRaWAN dispositivos WAN.

- Para obtener información acerca de la creación de grupos de multidifusión, consulte[Cree grupos de multidifusión y agregue dispositivos al grupo \(p. 1175\)](#).
- Para obtener información acerca de la creación de tareas de FUOTA, consulte[Cree una tarea FUOTA y proporcione una imagen de firmware \(p. 1186\)](#).

Cree grupos de multidifusión y agregue dispositivos al grupo

Puede crear grupos de multidifusión mediante la consola o la CLI. Si vas a crear tu grupo de multidifusión por primera vez, te recomendamos que uses la consola para agregar tu grupo de multidifusión. Cuando desee administrar su grupo de multidifusión y agregar o quitar dispositivos de su grupo, puede utilizar la CLI.

Después de intercambiar la señalización con los dispositivos finales que ha agregado,AWS IoT Core for LoRaWAN establece las claves compartidas con los dispositivos finales y establece los parámetros para la transferencia de datos.

Requisitos previos

Antes de crear grupos de multidifusión y agregar dispositivos al grupo:

- Prepare sus dispositivos para la configuración de multidifusión y FUOTA especificando los parámetros de configuración de FUOTAGenAppKeyyFPorts. Para obtener más información, consulte [Preparar dispositivos para la configuración de multidifusión y FUOTA \(p. 1173\)](#).
- Compruebe si los dispositivos admiten modos de operación clase B o clase C. En función de la clase de dispositivo que admite su dispositivo, elija un perfil de dispositivo que tenga uno o ambos. Soporta clase BoSoporta clase Cmodos habilitados. Para obtener información sobre los perfiles de dispositivos, consulte[Agregar perfiles a AWS IoT Core for LoRaWAN \(p. 1132\)](#).

Al inicio de la sesión de multidifusión, se utiliza una ventana de distribución de clase B o clase C para enviar mensajes de enlace descendente a los dispositivos del grupo.

Cree grupos de multidifusión mediante la consola

Para crear grupos de multidifusión mediante la consola de, vaya a la[Grupos de la multidifusión](#)Página de laAWS IoTconsola de y elijaCreación de un grupo de multidifusión.

1. Cree un grupo de multidifusión

Para crear un grupo de multidifusión, especifique las propiedades y etiquetas de multidifusión del grupo.

1. Especificación de propiedades de multidifusión

Para especificar las propiedades de multidifusión, introduzca la siguiente información para su grupo de multidifusión.

- Nombre: Escriba un nombre único para el grupo de multidifusión. El nombre solo debe incluir letras, números, guiones y guiones bajos. No puede contener espacios.
- Descripción: Puede proporcionar una descripción opcional para su grupo de multidifusión. La longitud de descripción puede tener hasta 2048 caracteres

2. Etiquetas para grupo de multidifusión

Opcionalmente, puede proporcionar cualquier par de clave/valor comoEtiquetaspara su grupo de multidifusión. Para seguir creando tu grupo de multidifusión, eligePróximo.

2. Agregue dispositivos a un grupo de multidifusión

Puede agregar dispositivos individuales o un grupo de dispositivos a su grupo de multidifusión. Para añadir dispositivos:

1. Especifique región

Especifique laRfRegiono banda de frecuencia para tu grupo de multidifusión. LaRfRegionpara que su grupo de multidifusión debe coincidir con laRfRegionde dispositivos que añada al grupo de multidifusión. Para obtener más información acerca de laRfRegion, consulte[Considerar la selección deLoRabandas de frecuencia para las puertas de enlace y la conexión de dispositivos \(p. 1124\)](#).

2. Seleccionar una clase de dispositivo de multidifusión

Elija si desea que los dispositivos del grupo de multidifusión cambien a un modo clase B o clase C al inicio de la sesión de multidifusión. Una sesión de clase B puede recibir mensajes de enlace descendente en ranuras de enlace descendente normales y una sesión de clase C puede recibir mensajes de enlace descendente en cualquier momento.

3. Elija los dispositivos que desea añadir al grupo

Elija si desea agregar dispositivos de forma individual o masiva al grupo de multidifusión.

- Para añadir dispositivos individualmente, introduzca el ID de dispositivo inalámbrico de cada dispositivo que deseé agregar a su grupo.
- Para añadir dispositivos de forma masiva, puedes filtrar los dispositivos que quieras agregar por perfil de dispositivo o etiquetas. En el caso del perfil de dispositivo, puede agregar dispositivos con un perfil compatible con clase B, clase C o ambas clases de dispositivo.

4. Para crear un grupo de multidifusión, elijaCrear.

Los detalles del grupo de multidifusión y los dispositivos que ha agregadoaparecer en el grupo. Para obtener información sobre el estado del grupo de multidifusión y de sus dispositivos y

para solucionar cualquier problema, consulte [Supervisar y solucionar problemas del grupo de multidifusión y de los dispositivos del grupo \(p. 1179\)](#).

Después de crear un grupo de multidifusión, puede elegir [Acción](#) para editar, eliminar o añadir dispositivos al grupo de multidifusión. Después de agregar los dispositivos, puedes programar una sesión para que la carga útil del enlace descendente se envíe a los dispositivos de tu grupo.

Crear grupos de multidifusión mediante la API

Para crear grupos de multidifusión y agregar dispositivos al grupo mediante la API:

1. Cree un grupo de multidifusión

Para crear un grupo de multidifusión, utilice la [CreateMulticastGroup](#) Operación de la API [create-multicast-group](#) Command de la CLI. Puede proporcionar un `input.json` archivo como entrada a la `create-multicast-group` comando.

```
aws iotwireless create-multicast-group \
--cli-input-json file://input.json
```

donde:

Contenido de `input.json`

```
{
  "Description": "Multicast group to send downlink payload and perform FUOTA updates.",
  "LoRaWAN": {
    "DLClass": "ClassB",
    "RfRegion": "US915"
  },
  "Name": "MC_group_FUOTA"
}
```

Después de crear el grupo de multidifusión, puede utilizar las siguientes operaciones de API o comandos de CLI para actualizar, eliminar u obtener información sobre los grupos de multidifusión.

- [UpdateMulticastGroup](#) o `update-multicast-group`
- [GetMulticastGroup](#) o `get-multicast-group`
- [ListMulticastGroups](#) o `list-multicast-groups`
- [DeleteMulticastGroup](#) o `delete-multicast-group`

2. Agregue dispositivos a un grupo de multidifusión

Puede agregar dispositivos a su grupo de multidifusión de forma individual o masiva.

- Para añadir dispositivos de forma masiva al grupo de multidifusión, utilice la [StartBulkAssociateWirelessDeviceWithMulticastGroup](#) Operación de la API [start-bulk-associate-wireless-device-with-multicast-group](#) Command de la CLI. Para filtrar los dispositivos que desea asociar de forma masiva a su grupo de multidifusión, proporcione una cadena de consulta. A continuación se muestra cómo agregar un grupo de dispositivos que tiene un perfil de dispositivo con el ID especificado vinculado a él.

```
aws iotwireless start-bulk-associate-wireless-device-with-multicast-group \
--id "12abd34e-5f67-89c2-9293-593b1bd862e0" \
--cli-input-json file://input.json
```

donde:

Contenido de input.json

```
{  
    "QueryString": "DeviceProfileName: MyWirelessDevice AND DeviceProfileId:  
d6d8ef8e-7045-496d-b3f4-ebcaa1d564bf",  
    "Tags": [  
        {  
            "Key": "Multicast",  
            "Value": "ClassB"  
        }  
    ]  
}
```

Aquí, `multicast-groups/d6d8ef8e-7045-496d-b3f4-ebcaa1d564bf/bulkes` la URL que se utiliza para asociar dispositivos al grupo.

- Para añadir dispositivos individualmente a su grupo de multidifusión, utilice la [AssociateWirelessDeviceWithMulticastGroup](#) Operación de la API de `associate-wireless-device-with-multicast-group` CLI. Proporciona el ID de dispositivo inalámbrico de cada dispositivo que quieras agregar a tu grupo.

```
aws iotwireless associate-wireless-device-with-multicast-group \  
--id "12abd34e-5f67-89c2-9293-593b1bd862e0" \  
--wireless-device-id "ab0c23d3-b001-45ef-6a01-2bc3de4f5333"
```

Después de crear el grupo de multidifusión, puede utilizar las siguientes operaciones de API o comandos de CLI para obtener información sobre el grupo de multidifusión o para disociar dispositivos.

- [DisassociateWirelessDeviceFromMulticastGroup](#) o `disassociate-wireless-device-from-multicast-group`
- [StartBulkDisassociateWirelessDeviceFromMulticastGroup](#) o `start-bulk-disassociate-wireless-device-from-multicast-group`
- [ListWirelessDevices](#) o `list-wireless-devices`

Note

La `ListWirelessDevices` El funcionamiento de la API se puede utilizar para enumerar dispositivos inalámbricos en general y dispositivos inalámbricos asociados a un grupo de multidifusión o a una tarea FUOTA.

- Para enumerar los dispositivos inalámbricos asociados a un grupo de multidifusión, utilice la `ListWirelessDevices` Operación API con `MulticastGroupId` como filtro.
- Para enumerar los dispositivos inalámbricos asociados a una tarea FUOTA, utilice la `ListWirelessDevices` Operación de la API con `FuotaTaskId` como filtro.

Pasos siguientes

Después de crear un grupo de multidifusión y agregar dispositivos, puede seguir añadiendo dispositivos y supervisar el estado del grupo de multidifusión y de sus dispositivos. Si los dispositivos se han agregado correctamente al grupo, puede configurar y programar un mensaje de enlace descendente para que se envíe a los dispositivos. Para poder enviar un mensaje de enlace descendente, el estado de sus dispositivos debe ser `Listo` para configurar la multidifusión. Después de programar un mensaje de enlace

descendente, el estado cambia al intento de sesión. Para obtener más información, consulte [Programar un mensaje de enlace descendente para enviarlo a los dispositivos del grupo de multidifusión \(p. 1181\)](#).

Si desea actualizar el firmware de los dispositivos del grupo de multidifusión, puede realizar actualizaciones de firmware por aire (FUOTA) con AWS IoT Core for LoRaWAN. Para obtener más información, consulte [Actualizaciones de firmware por aire \(FUOTA\) para AWS IoT Core for LoRaWAN Dispositivos de \(p. 1183\)](#).

Si no se han añadido tus dispositivos o si ves un error en el grupo de multidifusión o en los estados de dispositivos, puedes pasar el cursor sobre el error para obtener más información y resolverlo. Si sigue apareciendo un error, para obtener información sobre cómo solucionar el problema y resolver el problema, consulte [Supervisar y solucionar problemas del grupo de multidifusión y de los dispositivos del grupo \(p. 1179\)](#).

Supervisar y solucionar problemas del grupo de multidifusión y de los dispositivos del grupo

Después de agregar dispositivos y crear el grupo de multidifusión, abra el AWS Management Console. Vaya a la [.Grupos de la multidifusión](#) Página de la AWS IoT elija el grupo de multidifusión que creó para ver sus detalles. Verás información sobre el grupo de multidifusión, el número de dispositivos que se han agregado y los detalles de estado. Puede utilizar la información de estado para realizar un seguimiento del progreso de la sesión de multidifusión y solucionar cualquier error.

Estado del grupo de multidifusión

El grupo de multidifusión puede mostrar uno de los siguientes mensajes de estado en el AWS Management Console.

- Pendiente

Este estado indica que ha creado un grupo de multidifusión pero aún no tiene sesión de multidifusión. Aparecerá este mensaje de estado cuando se haya creado su grupo. Durante este tiempo, puede actualizar el grupo de multidifusión y asociar o disociar dispositivos con el grupo. Cuando el estado cambie de Pendiente, los dispositivos adicionales no se añadirán al grupo.

- Intento de sesión

Una vez que los dispositivos se hayan agregado correctamente al grupo de multidifusión, cuando el grupo tenga una sesión de multidifusión programada, verá que se muestra este mensaje de estado. Durante este tiempo, no puede actualizar o agregar dispositivos a su grupo de multidifusión. Si cancela la sesión de multidifusión, el estado del grupo cambia a Pendiente.

- En sesión

Cuando es el primer momento de la sesión para tu sesión de multidifusión, verás este mensaje de estado mostrado. Un grupo de multidifusión también continúa en este estado cuando está asociado a una tarea de FUOTA que tiene una sesión de actualización de firmware en curso.

Si no tiene una tarea FUOTA asociada en la sesión y si la sesión de multidifusión se cancela porque el tiempo de la sesión superó el tiempo de espera o canceló la sesión de multidifusión, el estado del grupo cambia a Pendiente.

- Eliminar espera

Si elimina el grupo de multidifusión, el estado de su grupo cambia a Eliminar espera. Las eliminaciones son permanentes y no se pueden deshacer. Esta acción puede llevar tiempo y el estado del grupo será Delete_Esperando hasta que se haya eliminado el grupo de multidifusión. Una vez que el grupo de multidifusión entre en este estado, no puede hacer la transición a uno de los otros estados.

Estado de los dispositivos del grupo de multidifusión

Los dispositivos del grupo de multidifusión pueden mostrar uno de los siguientes mensajes de estado en la AWS Management Console. Puede pasar el ratón sobre cada mensaje de estado para obtener más información sobre lo que indica.

- Intento de paquetes

Una vez que los dispositivos se hayan asociado al grupo de multidifusión, el estado del dispositivo esIntento de paquetes. Este estado indica que AWS IoT Core for LoRaWAN no ha confirmado si el dispositivo admite la configuración y el funcionamiento de multidifusión.

- Paquete no admitido

Una vez que los dispositivos se hayan asociado al grupo de multidifusión, AWS IoT Core for LoRaWAN comprueba si el firmware del dispositivo es capaz de configurar y operar multidifusión. Si el dispositivo no tiene el paquete de multidifusión compatible, su estado esPaquete no admitido. Para resolver el error, compruebe si el firmware del dispositivo es capaz de configurar y operar multidifusión.

- Intento de configuración de multidifusión

Si los dispositivos asociados a su grupo de multidifusión son capaces de configurar y operar multidifusión, el estado esIntento de configuración de multidifusión. Este estado indica que el dispositivo aún no ha completado la configuración de multidifusión.

- Listo para configurar la multidifusión

El dispositivo ha completado la configuración de multidifusión y se ha agregado al grupo de multidifusión. Este estado indica que los dispositivos están listos para una sesión de multidifusión y se puede enviar un mensaje de enlace descendente a esos dispositivos. El estado también indica cuándo puedes usar FUOTA para actualizar el firmware de los dispositivos del grupo.

- Intento de sesión

Se ha programado una sesión de multidifusión para los dispositivos del grupo de multidifusión. Al inicio de una sesión de grupo de multidifusión, el estado del dispositivo esIntento de sesión, y solicitudes se envían para saber si se puede iniciar una ventana de distribución de clase B o clase C para la sesión. Si el tiempo que tarda en configurar la sesión de multidifusión supera el tiempo de espera o si cancela la sesión de multidifusión, el estado cambia aConfiguración de la multidifusión.

- En sesión

Este estado indica que se ha iniciado una ventana de distribución de clase B o clase C y que el dispositivo tiene una sesión de multidifusión en curso. Durante este tiempo, los mensajes de enlace descendente se pueden enviar desde AWS IoT Core for LoRaWAN a los dispositivos del grupo de multidifusión. Si actualiza la hora de la sesión, anula la sesión actual y el estado cambia aIntento de sesión. Cuando finaliza la hora de la sesión o si cancela la sesión de multidifusión, el estado cambia aListo para configurar la multidifusión.

Pasos siguientes

Ahora que ha aprendido los distintos estados de su grupo de multidifusión y de los dispositivos de su grupo, y cómo puede solucionar problemas, como cuando un dispositivo no es capaz de configurar multidifusión, puede programar un mensaje de enlace descendente para que se envíe a los dispositivos y el grupo de multidifusión estará enEn sesión. Para obtener información acerca de la programación de un mensaje de enlace descendente, consulte[Programar un mensaje de enlace descendente para enviarlo a los dispositivos del grupo de multidifusión \(p. 1181\)](#).

Programar un mensaje de enlace descendente para enviarlo a los dispositivos del grupo de multidifusión

Después de agregar dispositivos correctamente al grupo de multidifusión, puede iniciar una sesión de multidifusión y configurar un mensaje de enlace descendente para que se envíe a esos dispositivos. El mensaje de enlace descendente debe estar programado en menos de 48 horas aY la hora de inicio de la multidifusión debe ser enmenos 30 minutos después de la hora actual.

Note

Los dispositivos de un grupo de multidifusión no pueden confirmar cuándo se ha recibido un mensaje de enlace descendente.

Requisitos previos

Para poder enviar un mensaje de enlace descendente, debe haber creado un grupo de multidifusión y haber agregado correctamente dispositivos al grupo para el que desea enviar un mensaje de enlace descendente. No puedes añadir más dispositivos después de programar una hora de inicio para tu sesión de multidifusión. Para obtener más información, consulte [Cree grupos de multidifusión y agregue dispositivos al grupo \(p. 1175\)](#).

Si alguno de los dispositivos no se ha añadido correctamente, el grupo de multidifusión y el estado del dispositivo contendrán información que le ayudará a resolver los errores. Si los errores persisten, para obtener información sobre la solución de problemas de estos errores, consulte [Supervisar y solucionar problemas del grupo de multidifusión y de los dispositivos del grupo \(p. 1179\)](#).

Programar un mensaje de enlace descendente mediante la consola

Para enviar un mensaje de enlace descendente mediante la consola de, vaya a la[Grupos de la multidifusión](#)Página de laAWS IoT elija el grupo de multidifusión que ha creado. En la página de detalles del grupo de multidifusión, elijaMensaje de enlace descendente dey, a continuación, elijaProgramar sesión de enlace descend.

1. Ventana de mensajes de enlace descendente

Puede configurar una ventana de tiempo para que se envíe un mensaje de enlace descendente a los dispositivos del grupo de multidifusión. El mensaje de enlace descendente debe programarse en un plazo de 48 horas.

Para programar la sesión de multidifusión, especifique los siguientes parámetros:

- Fecha de inicioyHora de inicio: La fecha y la hora de inicio deben ser de al menos 30 minutos después de y 48 horas antes de la hora actual.

Note

La hora que especifique está en UTC, así que considere comprobar la diferencia horaria con su zona horaria al programar la ventana de enlace descendente.

- Tiempo de espera de la sesión: Tiempo transcurrido el cual desea que se agote el tiempo de espera de la sesión de multidifusión si no se ha recibido ningún mensaje de enlace descendente. El tiempo de espera mínimo permitido es de 60 segundos. El valor máximo de tiempo de espera es de 2 días para los grupos de multidifusión de clase B y 18 horas para los grupos de multidifusión de clase C.

2. Configurar el mensaje de enlace descendente

Para configurar el mensaje de enlace descendente, especifique los siguientes parámetros:

- Rate de datos: Elige una velocidad de datos para tu mensaje de enlace descendente. La velocidad de datos depende de la región y del tamaño de la carga útil. La velocidad de datos predeterminada es 8 para la región US915 y 0 para la región EU868.

- Frecuencia: Elige una frecuencia para enviar el mensaje de enlace descendente. Para evitar conflictos de mensajería, elija una frecuencia disponible en función de la rfRegion.
- Fport: Elige un puerto de frecuencia disponible para enviar el mensaje de enlace descendente a tus dispositivos.
- Carga: Especifique el tamaño máximo de su carga útil en función de la velocidad de datos. Con la velocidad de datos predeterminada, puede tener un tamaño máximo de carga útil de 33 bytes en el US915RfRegion y 51 bytes en la EU868RfRegion. Con velocidades de datos más grandes, puede transferir hasta un tamaño máximo de carga útil de 242 bytes.

Para programar tu mensaje de enlace descendente, eligeSchedule.

Programar un mensaje de enlace descendente mediante la API

Para programar un mensaje de enlace descendente mediante la API, utilice la[StartMulticastGroupSession](#)Operación de la API de[start-multicast-group-session](#)Command de la CLI.

Puede utilizar las siguientes operaciones de API o comandos de CLI para obtener información sobre un grupo de multidifusión y eliminar un grupo de multidifusión.

- [GetMulticastGroupSession](#) o `get-multicast-group-session`
- [DeleteMulticastGroupSession](#) o `delete-multicast-group-session`

Para enviar datos a un grupo de multidifusión una vez iniciada la sesión, utilice el[SendDataToMulticastGroup](#)Operación de la API de[send-data-to-multicast-group](#)Command de la CLI.

Pasos siguientes

Después de configurar un mensaje de enlace descendente para enviarlo a los dispositivos, el mensaje se envía al inicio de la sesión. Los dispositivos de un grupo de multidifusión no pueden confirmar si se ha recibido el mensaje.

Configurar mensajes de enlace descendente adicionales

También puede configurar mensajes de enlace descendente adicionales para que se envíen a los dispositivos del grupo de multidifusión:

- Para configurar mensajes de enlace descendente adicionales desde la consola:
 1. Vaya a la[Grupos de la multidifusión](#)Página de la AWS IoT elija el grupo de multidifusión que ha creado.
 2. En la página de detalles del grupo de multidifusión, elijaMensaje de enlace descendente dey, a continuación, elijaConfigurar mensaje de enlace descendente adicional.
 3. Especificación de los parámetrosRate de datos,Frecuencia,Fport, yCarga, similar a cómo configuró estos parámetros para el primer mensaje de enlace descendente.
- Para configurar mensajes de enlace descendente adicionales mediante la API o la CLI, llame al[SendDataToMulticastGroup](#)Operación de la API de[send-data-to-multicast-group](#)Comando CLI para cada mensaje de enlace descendente adicional.

Actualizar la programación de sesiones

También puede actualizar la programación de sesiones para utilizar una nueva fecha y hora de inicio para la sesión de multidifusión. La nueva programación de sesiones anulará la sesión programada anteriormente.

Note

Actualice la sesión de multidifusión solo cuando sea necesario. Estas actualizaciones pueden provocar que un grupo de dispositivos se despierte durante mucho tiempo y agote la batería.

- Para actualizar la programación de sesiones desde la consola:
 1. Vaya a la[Grupos de la multidifusión](#)Página de la AWS IoT elija el grupo de multidifusión que ha creado.
 2. En la página de detalles del grupo de multidifusión, elijaMensaje de enlace descendente dey, a continuación, elijaActualizar la programación de sesiones.
 3. Especificación de los parámetrosFecha de estado,Hora de inicio, yTiempo de espera de la sesión, similar a cómo especificó estos parámetros para el primer mensaje de enlace descendente.
- Para actualizar la programación de sesiones desde la API o la CLI, utilice la[StartMulticastGroupSession](#)Operación de la API de[start-multicast-group-session](#)Command de la CLI.

Actualizaciones de firmware por aire (FUOTA) paraAWS IoT Core for LoRaWANDispositivos de

Usar actualizaciones de firmware por aire (FUOTA) para implementar actualizaciones de firmware enAWS IoT Core for LoRaWANDispositivos.

Con FUOTA, puedes enviar actualizaciones de firmware a dispositivos individuales o a un grupo de dispositivos. También puede enviar actualizaciones de firmware a varios dispositivos creando un grupo de multidifusión. En primer lugar, agregue los dispositivos al grupo de multidifusión y, a continuación, envíe la imagen de actualización de firmware a todos esos dispositivos. Le recomendamos que firme digitalmente las imágenes del firmware para que los dispositivos que reciban las imágenes puedan verificar que provienen de la fuente correcta.

conAWS IoT Core for LoRaWANlas actualizaciones de FUOTA, puedes:

- Implementar nuevas imágenes de firmware o imágenes delta en un único dispositivo o grupo de dispositivos.
- Verificar la autenticidad y la integridad del nuevo firmware una vez implementado en los dispositivos.
- Supervisar el progreso de una implementación y problemas de depuración en caso de que se produzca un error en la implementación.

AWS IoT Core for LoRaWANel soporte de FUOTA y grupos de multidifusión se basa en el[LoRaAlliance](#)especificaciones siguientes:

- LoRaEspecificación de configuración de multidifusión remota de WAN, TS005-2.0.0
- LoRaEspecificación de transporte de bloques de datos fragmentados WAN, TS004-2.0.0
- LoRaEspecificación de sincronización de reloj de capa de aplicaciones WAN, TS003-2.0.0

Note

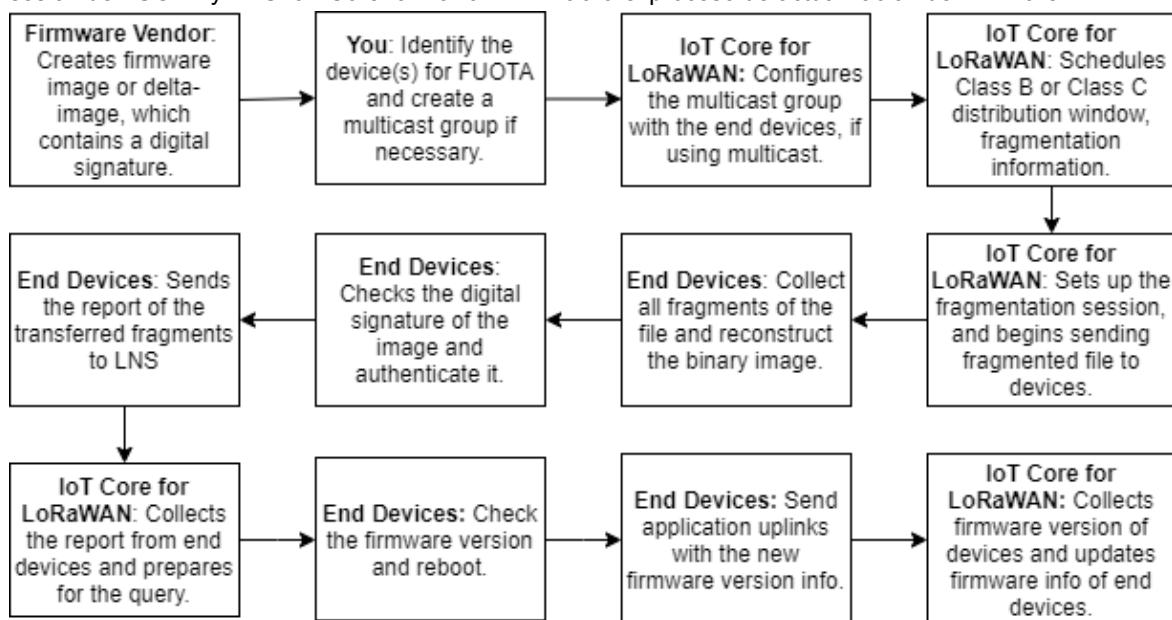
AWS IoT Core for LoRaWANrealiza automáticamente la sincronización del reloj según el[LoRaEspecificación de la alianza](#). Utiliza la función[AppTimeReq](#)para responder la hora del servidor a los dispositivos que lo solicitan mediante[ClockSync](#)señalización.

El siguiente ejemplo muestra cómo realizar actualizaciones de FUOTA.

- Información general del proceso FUOTA (p. 1184)
- Cree una tarea FUOTA y proporcione una imagen de firmware (p. 1186)
- Añadir dispositivos y grupos de multidifusión en una tarea FUOTA y programa una sesión de FUOTA (p. 1188)
- Monitorear solucionar problemas del estado de la tarea FUOTA y de los dispositivos agregados a la tarea (p. 1191)

Información general del proceso FUOTA

En el siguiente diagrama, se muestra cómo AWS IoT Core for LoRaWAN realiza el proceso FUOTA para sus dispositivos finales. Si vas a añadir dispositivos individuales a tu sesión de FUOTA, puedes omitir los pasos para crear y configurar tu grupo de multidifusión. Puedes añadir tus dispositivos directamente a una sesión de FUOTA y AWS IoT Core for LoRaWAN iniciará el proceso de actualización del firmware.



Para realizar actualizaciones de FUOTA para sus dispositivos, cree primero la imagen de firmware firmada digitalmente y configure los dispositivos y grupos de multidifusión que desea agregar a la tarea de FUOTA. Despues de iniciar una sesión de FUOTA, los dispositivos finales recopilan todos los fragmentos, reconstruyen la imagen de los fragmentos e informan del estado a AWS IoT Core for LoRaWAN, a continuación, aplique la nueva imagen de firmware.

A continuación se ilustran los diferentes pasos del proceso de FUOTA:

1. Cree una imagen de firmware o una imagen delta con una firma digital

Para AWS IoT Core for LoRaWAN para realizar actualizaciones de FUOTA para su LoRaWAN Dispositivos, le recomendamos que firme digitalmente la imagen del firmware o la imagen delta al enviar actualizaciones de firmware por aire. Los dispositivos que reciben las imágenes pueden comprobar que provienen de la fuente correcta.

La imagen del firmware no debe tener un tamaño superior a 1 megabyte. Cuanto mayor sea el tamaño del firmware, más tiempo puede tardar en completarse el proceso de actualización. Para una transferencia de datos más rápida o si la nueva imagen tiene más de 1 Megabyte, utilice una imagen delta, que es la parte de la nueva imagen que es el delta entre la nueva imagen de firmware y la imagen anterior.

Note

AWS IoT Core for LoRaWAN no proporciona la herramienta de generación de firmas digitales ni el sistema de administración de versiones de firmware. Puede utilizar cualquier herramienta de terceros para generar la firma digital de su imagen de firmware. Le recomendamos que utilice una herramienta de firma digital, como la incluida en el [Mbed con ARM GitHub repository](#), que también incluye herramientas para generar la imagen delta y para que los dispositivos utilicen esa imagen.

2. Identificar y configurar los dispositivos para FUOTA

Después de identificar los dispositivos de FUOTA, envíe actualizaciones de firmware a dispositivos individuales o a varios dispositivos.

- Para enviar las actualizaciones de firmware a varios dispositivos, cree un grupo de multidifusión y configure el grupo de multidifusión con dispositivos finales. Para obtener más información, consulte [Crear grupos de multidifusión para enviar una carga útil de enlace descendente a varios dispositivos \(p. 1172\)](#).
- Para enviar actualizaciones de firmware a dispositivos individuales, agréguelos a la sesión de FUOTA y, a continuación, realice la actualización del firmware.

3. Programar una ventana de distribución y configurar una sesión de fragmentación

Si ha creado un grupo de multidifusión, puede especificar la ventana de distribución de clase B o clase C para determinar cuándo los dispositivos pueden recibir los fragmentos de AWS IoT Core for LoRaWAN. Es posible que tus dispositivos estén funcionando en clase A antes de cambiar al modo clase B o clase C. También debe especificar la hora de inicio de la sesión.

Los dispositivos de clase B o clase C se activan en la ventana de distribución especificada y comienzan a recibir los paquetes de enlace descendente. Los dispositivos que funcionan en modo de clase C pueden consumir más energía que los dispositivos de clase B. Para obtener más información, consulte [Clases de dispositivo \(p. 1165\)](#).

4. Los dispositivos finales informan del estado a AWS IoT Core for LoRaWAN y actualizar la imagen del firmware

Después de configurar una sesión de fragmentación, los dispositivos finales y AWS IoT Core for LoRaWAN realice los siguientes pasos para actualizar el firmware de sus dispositivos.

1. PorqueLoRaLos dispositivos WAN tienen una baja velocidad de datos, para iniciar el proceso FUOTA,AWS IoT Core for LoRaWAN configura una sesión de fragmentación para fragmentar la imagen del firmware. Luegoenvía estos fragmentos a los dispositivos finales.
2. DespuésAWS IoT Core for LoRaWANenvía los fragmentos de imagen, suLoRaLos dispositivos finales WAN realizan las siguientes tareas.
 - a. Recoge los fragmentos y luego reconstruye la imagen binaria a partir de estos fragmentos.
 - b. Compruebe la firma digital de la imagen reconstruida para autenticar la imagen y verificar que proviene del origen correcto.
 - c. Compare la versión de firmware de AWS IoT Core for LoRaWAN a la versión actual.
 - d. Informar sobre el estado de las imágenes fragmentadas que se transfirieron a AWS IoT Core for LoRaWAN, a continuación, aplique la nueva imagen de firmware.

Note

En algunos casos,los dispositivos finales informan del estado de las imágenes fragmentadas que se transfirieron a AWS IoT Core for LoRaWANantes de comprobar la firma digital de la imagen del firmware.

Ahora que has aprendido el proceso de FUOTA, puedes crear tu tarea FUOTA y agregar dispositivos a la tarea para actualizar su firmware. Para obtener más información, consulte [Cree una tarea FUOTA y proporcione una imagen de firmware \(p. 1186\)](#).

Cree una tarea FUOTA y proporcione una imagen de firmware

Para actualizar el firmware de suLoRaDispositivos WAN, primero cree una tarea FUOTA y proporcione la imagen de firmware firmada digitalmente que desea utilizar para la actualización. A continuación, puede agregar sus dispositivos y grupos de multidifusión a la tarea y programar una sesión FUOTA. Cuando se inicie la sesión,AWS IoT Core for LoRaWANconfigura una sesión de fragmentación y los dispositivos finales recopilan los fragmentos, reconstruyen la imagen y aplican el nuevo firmware. Para obtener información acerca del proceso de FUOTA, consulte[Información general del proceso FUOTA \(p. 1184\)](#).

A continuación se muestra cómo crear una tarea FUOTA y cargar la imagen de firmware o la imagen delta que almacenará en un bucket de S3.

Requisitos previos

Antes de poder realizar actualizaciones de FUOTA, la imagen del firmware debe estar firmada digitalmente para que los dispositivos finales puedan verificar la autenticidad de la imagen al aplicar la imagen. Puede utilizar cualquier herramienta de terceros para generar la firma digital de su imagen de firmware. Le recomendamos que utilice una herramienta de firma digital, como la incluida en el[Mbed con ARMGitHubrepository](#), que también incluye herramientas para generar la imagen delta y para que los dispositivos utilicen esa imagen.

Cree una tarea de FUOTA y cargue imágenes de firmware mediante la consola

Para crear una tarea FUOTA y cargar la imagen de firmware mediante la consola, vaya a la[Tareas de FUOTA](#)pestaña de la consola y, a continuación, elijaCreación de tarea de FUOTA.

1. Creación de tarea de FUOTA

Para crear la tarea FUOTA, especifique las propiedades y etiquetas de la tarea.

1. Especificar las propiedades de tareas FUOTA

Para especificar las propiedades de la tarea FUOTA, introduzca la siguiente información para la tarea FUOTA.

- Nombre: Escriba un nombre único para la tarea de FUOTA. El nombre solo debe incluir letras, números, guiones y guiones bajos. No puede contener espacios.
- Descripción: Puede proporcionar una descripción opcional para su grupo de multidifusión. El campo de descripción puede tener hasta 2048 caracteres
- RfRegion: Establece la banda de frecuencia para tu tarea FUOTA. La banda de frecuencia debe coincidir con la que utilizó para aprovisionar dispositivos inalámbricos o grupos de multidifusión.

2. Etiquetas para tarea FUOTA

Opcionalmente, puede proporcionar cualquier par de clave/valor comoEtiquetaspara tu tarea FUOTA. Para seguir creando tu tarea, eligePróximo.

2. Subir imagen de firmware

Elija el archivo de imagen de firmware que desea utilizar para actualizar el firmware de los dispositivos que agregue a la tarea FUOTA. El archivo de imagen del firmware se almacena en un bucket de S3. Puedes proporcionarAWS IoT Core for LoRaWANlos permisos para obtener acceso a la imagen del firmware en su nombre. Le recomendamos que firme digitalmente las imágenes del firmware para que se verifique su autenticidad cuando se realice la actualización del firmware.

1. Elegir archivo de imagen de firmware

Puede cargar un nuevo archivo de imagen de firmware en un bucket de S3 o elegir una imagen existente que ya se haya cargado en un bucket de S3.

Note

El archivo de imagen del firmware no debe tener un tamaño superior a 1 megabyte. Cuanto mayor sea el tamaño del firmware, más tiempo puede tardar en completarse el proceso de actualización.

- Para utilizar una imagen existente, elija Seleccionar una imagen de firmware existente, elige Examinar S3 y, a continuación, seleccione el archivo de imagen de firmware que desee usar.

AWS IoT Core for LoRaWAN rellena la URL de S3, que es la ruta de acceso al archivo de imagen de firmware del bucket de S3. El formato de la ruta de es `s3://bucket_name/file_name`. Para ver el archivo en la [Amazon Simple Storage Service](#) consola de, elija Vista.

- Para cargar una nueva imagen de firmware.
 - a. Elegir Subir una nueva imagen de firmware y cargue la imagen de firmware. El archivo de imagen no debe ser superior a 1 megabyte.
 - b. Para crear un bucket de S3 e introducir un nombre del bucket para almacenar el archivo de imagen de firmware, elija Create bucket de S3.

2. Permisos para obtener acceso al bucket de

Puede crear un rol de servicio nuevo o elegir un rol existente para permitir AWS IoT Core for LoRaWAN para obtener acceso al archivo de imagen de firmware en el bucket de S3 en su nombre. Elija Next (Siguiente).

Para crear un nuevo rol, puede introducir un nombre de rol o dejarlo en blanco para que se genere automáticamente un nombre aleatorio. Para ver los permisos de política que otorgan acceso al bucket de S3, seleccione Ver permisos de políticas.

Para obtener más información sobre el uso de un bucket de S3 para almacenar su imagen y conceder AWS IoT Core for LoRaWAN permisos para acceder a él, consulte [Cargue el archivo de firmware en un bucket de S3 y agregue una función de IAM \(p. 1157\)](#).

3. Revisar y crear

Para crear tu tarea FUOTA, revisa la tarea FUOTA y los detalles de configuración que especificaste y elige Create task (Crear tarea).

Cree una tarea FUOTA y cargue una imagen de firmware mediante la API

Para crear una tarea FUOTA y especificar el archivo de imagen de firmware mediante la API, utilice la `CreateFuotaTask` Operación de la API de `create-fuota-task` Command de la CLI. Puede proporcionar un `input.json` archivo como entrada a la `create-fuota-task` comando. Cuando utiliza la API o la CLI, el archivo de imagen de firmware que proporciona como entrada debe cargarse ya en un bucket de S3. También se especifica el rol de IAM que proporciona AWS IoT Core for LoRaWAN acceso a la imagen de firmware en el bucket de S3.

```
aws iotwireless create-fuota-task \
--cli-input-json file://input.json
```

donde:

Contenido de `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"
}
```

Después de crear la tarea FUOTA, puede utilizar las siguientes operaciones de API o comandos de CLI para actualizar, eliminar u obtener información sobre la tarea de FUOTA.

- [UpdateFuotaTask](#) o `update-fuota-task`
- [GetFuotaTask](#) o `get-fuota-task`
- [ListFuotaTasks](#) o `list-fuota-tasks`
- [DeleteFuotaTask](#) o `delete-fuota-task`

Pasos siguientes

Ahora que ha creado una tarea FUOTA y ha proporcionado la imagen del firmware, puede agregar dispositivos a la tarea para actualizar su firmware. Puede agregar dispositivos individuales o grupos de multidifusión a la tarea. Para obtener más información, consulte [Añadir dispositivos y grupos de multidifusión en una tarea FUOTA y programa una sesión de FUOTA](#) (p. 1188).

Añadir dispositivos y grupos de multidifusión en una tarea FUOTA y programa una sesión de FUOTA

Después de crear una tarea FUOTA, puede agregar dispositivos a la tarea para los que desea actualizar el firmware. Una vez que los dispositivos se hayan agregado correctamente a la tarea FUOTA, puedes programar una sesión de FUOTA para actualizar el firmware del dispositivo.

- Si solo tienes un pequeño número de dispositivos, puedes añadirlos directamente a tu tarea de FUOTA.
- Si tiene un gran número de dispositivos para el que desea actualizar firmware para, puede agregar estos dispositivos a los grupos de multidifusión y, a continuación, agregar los grupos de multidifusión a la tarea de FUOTA. Para obtener información acerca de cómo crear y usar grupos de multidifusión, consulte [Crear grupos de multidifusión para enviar una carga útil de enlace descendente a varios dispositivos](#) (p. 1172).

Note

Puede agregar dispositivos individuales o grupos de multidifusión a la tarea FUOTA. No puedes agregar dispositivos y grupos de multidifusión a la tarea.

Después de agregar los dispositivos o grupos de multidifusión, puede iniciar una sesión de actualización de firmware AWS IoT Core for LoRaWAN. Recopila la imagen del firmware, fragmenta las imágenes y, a continuación, almacena los fragmentos en un formato cifrado. Los dispositivos finales recopilan los fragmentos y aplican la nueva imagen de firmware. El tiempo que tarda la actualización del firmware depende del tamaño de la imagen y de cómo se fragmentaron las imágenes. Una vez finalizada la actualización del firmware, los fragmentos cifrados de la imagen del firmware almacenados por AWS IoT Core for LoRaWAN se eliminan. Aún puede encontrar la imagen del firmware en el bucket de S3.

Requisitos previos

Antes de agregar dispositivos o grupos de multidifusión a la tarea de FUOTA, haga lo siguiente.

- Debe de haber creado la tarea FUOTA y proporcionar su imagen de firmware. Para obtener más información, consulte [Cree una tarea FUOTA y proporcione una imagen de firmware](#) (p. 1186).

- Aprovisione los dispositivos inalámbricos para los que desea actualizar el firmware del dispositivo. Para obtener más información acerca de la incorporación de su dispositivo, consulte [Incorporación de sus dispositivos a AWS IoT Core for LoRaWAN \(p. 1129\)](#).
- Para actualizar el firmware de varios dispositivos, puede agregarlo a un grupo de multidifusión. Para obtener más información, consulte [Crear grupos de multidifusión para enviar una carga útil de enlace descendente a varios dispositivos \(p. 1172\)](#).
- Cuando incorporas los dispositivos a AWS IoT Core for LoRaWAN, especifique el parámetro de configuración FUOTAFFPorts. Si utiliza unLoRaWAN v1.0.x, también debe especificar laGenAppKey. Para obtener más información acerca de los parámetros de configuración de FUOTA, consulte [Preparar dispositivos para la configuración de multidifusión y FUOTA \(p. 1173\)](#).

Agregar dispositivos a una tarea de FUOTA y programar una sesión de FUOTA mediante la consola

Para agregar dispositivos o grupos de multidifusión y programar una sesión FUOTA mediante la consola, vaya a la sección [Tareas de FUOTA](#) pestaña de la consola. A continuación, elija la tarea FUOTA a la que desea agregar dispositivos y realice la actualización del firmware.

Agregar dispositivos y grupos de multidifusión

1. Puede agregar dispositivos individuales o grupos de multidifusión a la tarea de FUOTA. Sin embargo, no puedes agregar dispositivos individuales y grupos de multidifusión a la misma tarea de FUOTA. Para agregar dispositivos mediante la consola de, haga lo siguiente.
 1. En el navegador [Detalles de tareas de FUOTA](#), elige [Añadir dispositivo](#).
 2. Elija la banda de frecuencia o [RfRegion](#) para los dispositivos que añadas a la tarea. Este valor debe coincidir con la [RfRegion](#) que elegiste para la tarea FUOTA.
 3. Elija si desea agregar dispositivos individuales o grupos de multidifusión a la tarea.
 - Para añadir dispositivos individuales, elija [Añadir dispositivos individuales](#) e introduce el ID de dispositivo de cada dispositivo que quieras agregar a tu tarea de FUOTA.
 - Para añadir grupos de multidifusión, elija [Añadir grupos de multidifusión](#) y añada los grupos de multidifusión a la tarea. Puede filtrar los grupos de multidifusión que desea agregar a la tarea mediante el perfil o las etiquetas del dispositivo. Al filtrar por perfil de dispositivo, puede elegir grupos de multidifusión que tienen un perfil con [Soporta clase Bo](#) o [Soporta clase Ch](#) habilitado.
2. Programar sesión de FUOTA

Una vez que los dispositivos o grupos de multidifusión se hayan agregado correctamente, puede programar una sesión FUOTA. Para programar una sesión, haga lo siguiente.

1. Seleccione la tarea de FUOTA para la que desea actualizar el firmware del dispositivo y, a continuación, seleccione [Programar sesión de FUOTA](#).
2. Especificar un [Fecha de inicio](#) y [Hora de inicio](#) para tu sesión FUOTA. Asegúrese de que la hora de inicio sea 30 minutos o más tarde desde la hora actual.

Agregar dispositivos a una tarea de FUOTA y programar una sesión de FUOTA mediante la API

Puede utilizar el [AWS IoT Wireless API](#) o la CLI para agregar dispositivos inalámbricos o grupos de multidifusión a la tarea de FUOTA. A continuación, puede programar una sesión de FUOTA.

1. Agregar dispositivos y grupos de multidifusión

Puede asociar dispositivos inalámbricos o grupos de multidifusión a su tarea FUOTA.

- Para asociar dispositivos individuales a la tarea de FUOTA, utilice el [AssociateWirelessDeviceWithFuotaTask](#) Operación de la API de `associate-wireless-device-with-fuota-task` comando CLI y proporcione el `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 asociar grupos de multidifusión a la tarea de FUOTA, utilice la [AssociateMulticastGroupWithFuotaTask](#) Operación de la API de `associate-multicast-group-with-fuota-task` comando CLI y proporcione el `MulticastGroupID` como entrada.

```
aws iotwireless associate-multicast-group-with-FUOTA-task \
--id 01a23cde-5678-4a5b-ab1d-33456808ecb2"
--multicast-group-id
```

Después de asociar los dispositivos inalámbricos o el grupo de multidifusión a la tarea de FUOTA, utilice las siguientes operaciones de API o comandos de CLI para enumerar los dispositivos o grupos de multidifusión o para desasociarlos de la tarea.

- [DisassociateWirelessDeviceFromFuotaTask](#) o `disassociate-wireless-device-from-fuota-task`
- [DisassociateMulticastGroupFromFuotaTask](#) o `disassociate-multicast-group-from-fuota-task`
- [ListWirelessDevices](#) o `list-wireless-devices`
- [ListMulticastGroups](#) o `list-multicast-groups-by-fuota-task`

Note

La API:

- `ListWirelessDevices` puede enumerar dispositivos inalámbricos en general y dispositivos asociados a un grupo de multidifusión, cuando `MulticastGroupID` se utiliza como filtro. La API enumera los dispositivos inalámbricos asociados a una tarea FUOTA cuando `FuotaTaskID` se utiliza como filtro.
- `ListMulticastGroups` puede enumerar grupos de multidifusión en general y grupos de multidifusión asociados a una tarea FUOTA cuando `FuotaTaskID` se utiliza como filtro.

2. Programar sesión de FUOTA

Una vez que los dispositivos o grupos de multidifusión se hayan agregado correctamente a la tarea FUOTA, puede iniciar una sesión de FUOTA para actualizar el firmware del dispositivo. La hora de inicio debe ser 30 minutos o más tarde a partir de la hora actual. Para programar una sesión de FUOTA mediante la API o la CLI, utilice la [StartFuotaTask](#) Operación de la API de `start-fuota-task` Command de la CLI.

Después de iniciar una sesión de FUOTA, ya no podrá agregar dispositivos o grupos de multidifusión a la tarea. Puede obtener información sobre el estado de su sesión de FUOTA mediante el [GetFuotaTask](#) Operación de la API de `get-fuota-task` Command de la CLI.

Monitory solucionar problemas del estado de la tarea FUOTA y de los dispositivos agregados a la tarea

Después de aprovisionar los dispositivos inalámbricos y crear los grupos de multidifusión que deseé utilizar, puede iniciar una sesión FUOTA siguiendo los siguientes pasos.

Estado de tarea de FUOTA

La tarea de FUOTA puede mostrar uno de los siguientes mensajes de estado en el AWS Management Console.

- Pendiente

Este estado indica que has creado una tarea FUOTA, pero aún no tiene una sesión de actualización de firmware. Aparecerá este mensaje de estado cuando se haya creado la tarea. Durante este tiempo, puede actualizar su tarea de FUOTA y asociarlo disociar dispositivos o grupos de multidifusión con la tarea. Cuando el estado cambie de Pendiente, no se pueden añadir dispositivos adicionales a la tarea.

- Sesión FUOTA en espera

Después de que tus dispositivos hayan sido conectados con éxito a la tarea FUOTA, cuando la tarea tenga una sesión de actualización de firmware programada, verá que se muestra este mensaje de estado. Durante este tiempo, no puede actualizar o añadir dispositivos a su sesión de FUOTA. Si cancelas tu sesión de FUOTA, el estado del grupo cambia a Pendiente.

- En la sesión de FUOTA

Cuando comience la sesión de FUOTA, aparecerá este mensaje de estado. Se inicia la sesión de fragmentación y los dispositivos finales recopilan los fragmentos, reconstruyen la imagen del firmware, comparan la nueva versión de firmware con la versión original y aplican la nueva imagen.

- FUOTA se ha hecho

Después de que los dispositivos finales informen a AWS IoT Core for LoRaWAN que se ha aplicado la nueva imagen del firmware o, cuando se agota el tiempo de espera de la sesión, la sesión FUOTA se marca como terminada y verá este estado en pantalla.

También verás este estado en cualquiera de los siguientes casos, así que asegúrate de comprobar si la actualización de firmware se ha aplicado correctamente a los dispositivos.

- Cuándo estaba el estado de la tarea FUOTASesión FUOTA en espera, y hay un error de bucket de S3, como el enlace al archivo de imagen en el bucket de S3 es incorrecto o AWS IoT Core for LoRaWAN no tiene permisos suficientes para obtener acceso al archivo del bucket.
- Cuándo estaba el estado de la tarea FUOTASesión FUOTA en espera, y hay una solicitud para iniciar una sesión de FUOTA, pero no se recibe ninguna respuesta de los dispositivos o grupos de multidifusión de la tarea de FUOTA.
- Cuándo estaba el estado de la tarea FUOTAEEn la sesión de FUOTA, y los dispositivos o grupos de multidifusión no han enviado fragmentos durante un período de tiempo determinado, lo que hace que la sesión se agote el tiempo de espera.
- Eliminar espera

Si eliminas tu tarea FUOTA que se encuentra en cualquiera de los demás estados, verás este estado mostrado. Una acción de eliminación es permanente y no se puede deshacer. Esta acción puede llevar tiempo y el estado de la tarea será Eliminar esperar hasta que se haya eliminado la tarea de FUOTA. Una vez que tu tarea de FUOTA entre en este estado, no puede hacer la transición a uno de los otros estados.

Estado de los dispositivos de una tarea FUOTA

Los dispositivos de la tarea FUOTA pueden mostrar uno de los siguientes mensajes de estado en el AWS Management Console. Puede pasar el ratón sobre cada mensaje de estado para obtener más información sobre lo que indica.

- Inicial

Cuando sea la hora de inicio de su sesión de FUOTA, AWS IoT Core for LoRaWAN comprueba si el dispositivo tiene el paquete compatible para la actualización del firmware. Si el dispositivo tiene el paquete compatible, se inicia la sesión FUOTA del dispositivo. La imagen del firmware está fragmentada y los fragmentos se envían al dispositivo. Cuando aparezca este estado, indica que la sesión FUOTA del dispositivo no se ha iniciado todavía.

- Paquete no admitido

Si el dispositivo no tiene el paquete FUOTA compatible, verás este estado en pantalla. Si el paquete de actualización de firmware no es compatible, la sesión FUOTA del dispositivo no se puede iniciar. Para resolver este error, comprueba si el firmware de tu dispositivo puede recibir actualizaciones de firmware mediante FUOTA.

- Algoritmo de fragmentación no admitido

Al comienzo de tu sesión de FUOTA, AWS IoT Core for LoRaWAN configura una sesión de fragmentación para el dispositivo. Si ves que se muestra este estado, significa que el tipo de algoritmo de fragmentación utilizado no se puede aplicar a la actualización del firmware de tu dispositivo. El error se produce porque el dispositivo no tiene el paquete FUOTA compatible. Para resolver este error, comprueba si el firmware de tu dispositivo puede recibir actualizaciones de firmware mediante FUOTA.

- No hay memoria suficiente

Después AWS IoT Core for LoRaWAN envía los fragmentos de imagen, los dispositivos finales recopilan los fragmentos de imagen y reconstruyen la imagen binaria a partir de estos fragmentos. Este estado se muestra cuando el dispositivo no tiene suficiente memoria para ensamblar los fragmentos entrantes de la imagen del firmware, lo que puede provocar que la sesión de actualización de firmware finalice prematuramente. Para resolver el error, comprueba si el hardware del dispositivo puede recibir esta actualización. Si tu dispositivo no puede recibir esta actualización, usa una imagen delta para actualizar el firmware.

- Índice de fragmentación no admitido

El índice de fragmentación identifica una de las cuatro sesiones de fragmentación posibles simultáneamente. Si el dispositivo no admite el valor de índice de fragmentación indicado, se muestra este estado. Para solucionar este error, realice una o varias de las acciones siguientes.

- Inicia una nueva tarea de FUOTA para el dispositivo.
- Si el error persiste, cambie del modo de unidifusión al modo de multidifusión.
- Si el error sigue sin resolverse, comprueba el firmware del dispositivo.
- Error de memoria

Este estado indica que el dispositivo ha experimentado un error de memoria al recibir los fragmentos entrantes de AWS IoT Core for LoRaWAN. Si se produce este error, es posible que el dispositivo no pueda recibir esta actualización. Para resolver el error, comprueba si el hardware del dispositivo puede recibir esta actualización. Si es necesario, utilice una imagen delta para actualizar el firmware del dispositivo.

- Descriptor erróneo

El dispositivo no admite el descriptor indicado. El descriptor es un campo que describe el archivo que se transportará durante la sesión de fragmentación. Si aparece este error, póngase en contacto con [AWS Support Centro](#).

- Reproducción de recuento de sesiones

Este estado indica que el dispositivo ha utilizado anteriormente este recuento de sesiones. Para solucionar el error, inicie una nueva tarea de FUOTA para el dispositivo.

- Fragmentos faltantes

A medida que el dispositivo recopila los fragmentos de imagen de AWS IoT for LoRaWAN, reconstruye la nueva imagen de firmware a partir de fragmentos codificados independientes. Si tu dispositivo no ha recibido todos los fragmentos, la nueva imagen no se puede reconstruir y verás este estado. Para solucionar el error, inicie una nueva tarea de FUOTA para el dispositivo.

- Error MIC

Cuando el dispositivo reconstruye la nueva imagen de firmware a partir de los fragmentos recopilados, realiza un MIC (Message Integrity Check) para verificar la autenticidad de la imagen y si procede del origen correcto. Si el dispositivo detecta una discrepancia en el MIC tras volver a ensamblar los fragmentos, se muestra este estado. Para solucionar el error, inicie una nueva tarea de FUOTA para el dispositivo.

- Exitoso

La sesión FUOTA de tu dispositivo se ha realizado correctamente.

Note

Aunque este mensaje de estado indica que los dispositivos han reconstruido la imagen a partir de los fragmentos y la han verificado, es posible que el firmware del dispositivo no se haya actualizado cuando el dispositivo informa del estado a AWS IoT for LoRaWAN. Compruebe si el firmware del dispositivo se ha actualizado.

Pasos siguientes

Has aprendido sobre los diferentes estados de la tarea FUOTA y sus dispositivos y cómo solucionar cualquier problema. Para obtener más información acerca de cada uno de estos estados, consulte la [Especificación de transporte de bloques de datos fragmentados WAN, TS004-1.0.0](#).

Supervisión de su flota de recursos inalámbricos en tiempo real mediante el analizador de redes

El analizador de red utiliza un valor predeterminado WebSocket conexión para recibir registros de mensajes de seguimiento en tiempo real para sus recursos de conectividad inalámbrica. Mediante el analizador de red, puede agregar los recursos que desea supervisar, activar una sesión de mensajería de seguimiento y comenzar a recibir mensajes de seguimiento en tiempo real.

Para monitorizar sus recursos, también puede utilizar Amazon CloudWatch. Para utilizar CloudWatch, configura un rol de IAM para configurar el registro y, a continuación, esperar a que las entradas de registro se muestren en la consola. El analizador de red reduce significativamente el tiempo que tarda en configurar una conexión y comenzar a recibir mensajes de seguimiento, proporcionándole just-in-time información de registro de su flota de recursos. Para obtener más información sobre la monitorización mediante CloudWatch, consulte [Monitoreo y registro de AWS IoT Wireless usando de Amazon CloudWatch \(p. 1220\)](#).

Al reducir el tiempo de configuración y utilizar la información de los mensajes de seguimiento, puede supervisar los recursos de forma más eficaz, obtener información significativa y solucionar errores. Puede supervisar los dispositivos LoraWAN y las puertas de enlace LoraWAN. Por ejemplo, puede identificar rápidamente un error de unión al incorporar uno de sus dispositivos LoraWAN. Para depurar el error, utilice la información del registro de mensajes de seguimiento proporcionado.

Cómo utilizar el analizador de red

Para supervisar la flota de recursos y comenzar a recibir mensajes de seguimiento, lleve a cabo los siguientes pasos

1. Cree la configuración del analizador de red y agregue recursos

Antes de activar la mensajería de seguimiento, cree una configuración de analizador de red y agregue recursos a la configuración. En primer lugar, especifique los ajustes de configuración, que incluyen niveles de registro e información de tramas de dispositivos inalámbricos. A continuación, agregue los recursos inalámbricos que desea supervisar mediante la puerta de enlace inalámbrica y los identificadores de dispositivos inalámbricos.

2. Transmitir mensajes de seguimiento con WebSockets

Puede generar una URL de solicitud prefirmada utilizando las credenciales de su rol de IAM para transmitir mensajes de seguimiento del analizador de red mediante la WebSocket protocolo.

3. Activar sesión de mensajería de seguimiento y supervisar los mensajes de seguimiento

Para empezar a recibir mensajes de seguimiento, active la sesión de mensajería de seguimiento. Para evitar costes adicionales, puede desactivar o cerrar la sesión de mensajería de seguimiento del analizador de red.

A continuación se muestra cómo crear la configuración, agregar recursos y activar la sesión de mensajería de seguimiento.

Temas

- [Agregar el rol de IAM necesario para el analizador de redes \(p. 1194\)](#)
- [Cree una configuración de analizador de red y agregue recursos \(p. 1196\)](#)
- [Transmitir mensajes de seguimiento del analizador de red con WebSockets \(p. 1202\)](#)
- [Ver y supervisar los registros de mensajes de seguimiento del analizador de red en tiempo real \(p. 1207\)](#)

Agregar el rol de IAM necesario para el analizador de redes

Cuando utiliza el analizador de red, debe conceder permiso a un usuario para utilizar las operaciones de la API [Actualización de la configuración de Network Analyzer](#) y [Obtener configuración de Network Analyzer](#) para acceder a los recursos del analizador de redes. A continuación se muestran las políticas de IAM que utiliza para conceder permisos.

Políticas de IAM para analizador de redes

Utilice una de las siguientes operaciones:

- Política inalámbrica de acceso completo

Conceder AWS IoT Core para LoRaWan la política de acceso completo adjuntando la política AWSIoTWirelessFullAccess a tu rol. Para obtener más información, consulte [AWSIoTWirelessFullAccess Resumen de políticas](#).

- Política de IAM de ámbito para obtener y actualizar API

Cree la siguiente política de IAM yendo a la [Crear política](#) de la consola de IAM y en la Visual editor (Editor visual) pestaña:

1. ElegerInalámbrico IoT para Service (Servicio).
2. UnderNivel de acceso, expandaLectura y eligeObtener configuración de Network Analyzer, a continuación, expandaEscritura y eligeActualización de la configuración de Network Analyzer.
3. ElegirSiguiente: Tagse introduzca unNombrepara la política, comoPolítica del analizador de redes inalámbricas de IoT. Elija Create Policy (Crear política).

A continuación se muestra la políticaPolítica del analizador de redes inalámbricas de IoT que creaste. Para obtener más información acerca de la creación de una política de, consulte[Creación de políticas de IAM](#).

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "VisualEditor0",  
            "Effect": "Allow",  
            "Action": [  
                "iotwireless:GetNetworkAnalyzerConfiguration",  
                "iotwireless:UpdateNetworkAnalyzerConfiguration"  
            ],  
            "Resource": "*"  
        }  
    ]  
}
```

Política de ámbito para acceder a recursos específicos

Para configurar un control de acceso más preciso, debe agregar las puertas de enlace y dispositivos inalámbricos a laRecurso. La siguiente política utiliza el ARN comodín para conceder acceso a todas las puertas de enlace y dispositivos. Puede controlar el acceso a puertas de enlace y dispositivos específicos medianteWirelessGatewayIdyWirelessDeviceId.

```
{  
    "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 permiso a un usuario para utilizar el analizador de redes pero no utilizar puertas de enlace o dispositivos inalámbricos, utilice la siguiente directiva. A menos que se especifique, se deniegan implícitamente los permisos para utilizar los recursos.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "VisualEditor0",  
            "Effect": "Deny",  
            "Action": [  
                "iotwireless:UpdateNetworkAnalyzerConfiguration"  
            ],  
            "Resource": "*"  
        }  
    ]  
}
```

```
"Sid": "VisualEditor0",
"Effect": "Allow",
>Action": [
    "iotwireless:GetNetworkAnalyzerConfiguration",
    "iotwireless:UpdateNetworkAnalyzerConfiguration"
],
"Resource": [
    "arn:aws:iotwireless:*:{accountId}:NetworkAnalyzerConfiguration/*"
]
}
```

Pasos siguientes

Ahora que ha creado la política, puede agregar recursos a la configuración del analizador de red y recibir información de mensajería de seguimiento para esos recursos. Para obtener más información, consulte [Cree una configuración de analizador de red y agregue recursos \(p. 1196\)](#).

Cree una configuración de analizador de red y agregue recursos

Antes de poder transmitir mensajes de seguimiento, cree una configuración de analizador de red y agregue a esta configuración los recursos que desea supervisar. Al crear una configuración, puede:

- Especifique un nombre de configuración y una descripción opcional.
- Personalice los ajustes de configuración, como la información de fotogramas y el nivel de detalle de los mensajes de registro.
- Agregue los recursos que desea monitorear. Los recursos pueden ser dispositivos inalámbricos o puertas de enlace inalámbricas, o ambos.

La configuración que especifique determinará la información de mensajería de seguimiento que recibirá de los recursos que agregue a la configuración. También puede que desee crear varias configuraciones según el caso de uso de la supervisión.

A continuación, se muestra cómo crear una configuración y agregar recursos.

Temas

- [Creación de una configuración de analizador de red \(p. 1196\)](#)
- [Agregue recursos y actualice la configuración del analizador de red \(p. 1200\)](#)

Creación de una configuración de analizador de red

Antes de poder supervisar las puertas de enlace inalámbricas o los dispositivos inalámbricos, debe crear una configuración de analizador de red. Al crear la configuración, solo tiene que especificar un nombre de configuración. Puede personalizar los ajustes de configuración y agregar los recursos que desea supervisar a la configuración incluso después de crearla. La configuración determina la información de mensajería de seguimiento que recibirá para esos recursos.

En función de los recursos que desee supervisar y del nivel de información que desee recibir para ellos, puede que desee crear varias configuraciones. Por ejemplo, puede crear una configuración que muestre solo información de error para un conjunto de puertas de enlace de su cuenta de AWS. También puede crear una configuración que muestre toda la información sobre un dispositivo inalámbrico que desea supervisar.

En las secciones siguientes se muestran los diversos ajustes de configuración y cómo crear la configuración.

Opciones de configuración

Al crear o actualizar la configuración del analizador de red, también puede personalizar los siguientes parámetros para filtrar la información del flujo de registro.

- Información de marco

Esta configuración es la información de fotogramas de los recursos de su dispositivo inalámbrico para los mensajes de seguimiento. La información de trama se puede utilizar para depurar la comunicación entre el servidor de red y los dispositivos finales. Está habilitado de forma predeterminada.

- Niveles de registro

Puedes ver los registros de información o errores, o desactivar el registro.

- Información

Registros con un nivel de registro deErrorson más detallados y contienen flujos de registro de errores y flujos de registro informativos. Los registros informativos se pueden utilizar para ver los cambios en el estado de un dispositivo o puerta de enlace.

Note

La recopilación de flujos de registro más detallados puede suponer costes adicionales. Para obtener más información acerca de los precios, consulte [Precios de AWS IoT Core](#).

- Error

Registros con un nivel de registro deErrorson menos detallados y solo muestran información de error. Puede utilizar estos registros cuando una aplicación tiene un error, como un error de conexión del dispositivo. Al utilizar la información de la secuencia de registros, puede identificar y solucionar los errores de los recursos de su flota.

Creación de una configuración con la consola

Puede crear una configuración de analizador de red y personalizar los parámetros opcionales mediante laAWS IoTconsola deAWS IoTAPI inalámbrica. También puede crear varias configuraciones y, posteriormente, eliminar cualquier configuración que ya no utilice.

Creación de una configuración de analizador de red

1. Abra el icono [Network Analyzer hub](#) deAWS IoTconsola y eligeCreación de configuración.

2. Especifique los ajustes de configuración.

- Nombre, descripción y etiquetas

Especificar un únicoNombre de la configuraciónthatsolo contiene letras, números, guiones o guiones bajos. Utilice la opción opcionalDescripciónpara proporcionar información sobre la configuración yEtiquetaspara añadir pares clave-valor de metadatos sobre la configuración. Para obtener más información sobre cómo nombrar y describir los recursos de, consulte[Describa susAWS IoT CoreparaLoRaRecursos WAN \(p. 1121\)](#).

- Opciones de configuración

Elija si desea deshabilitar la información de fotogramas y utilizarSeleccione niveles de registrospara elegir los niveles de registro que desea utilizar para los registros de mensajes de seguimiento. Elija [Next \(Siguiente\)](#).

3. Agregue recursos a la configuración. Puedes añadir tus recursos ahora o elegir Crear, a continuación, añada los recursos más adelante. Para añadir recursos más adelante, elija Crear.

En el navegador Página del hub de Network Analyzer, verá la configuración que creó junto con sus ajustes. Para ver los detalles de la nueva configuración, seleccione el nombre de la configuración.

Eliminar la configuración del analizador de red

Puede crear varias configuraciones de analizador de red en función de los recursos que desee supervisar y del nivel de información de mensajería de seguimiento que desee recibir para ellos.

Para quitar configuraciones de la consola

1. Vaya a [Network Analyzer hub de AWS IoT](#) consola y elija la configuración que desea eliminar.
2. Elija Acciones y, a continuación, elija Eliminar.

Creación de una configuración utilizando la API

Para crear una configuración de analizador de red mediante la API, utilice la [Crear configuración de Network Analyzer](#) Operación de la API `create-network-analyzer-configuration` comando de la CLI.

Cuando crea la configuración, solo tiene que especificar un nombre de configuración. También puede utilizar esta operación de API para especificar los ajustes de configuración y agregar recursos al crear la configuración. Si lo desea, puede especificarlos más adelante utilizando la [Actualización de la configuración de Network Analyzer](#) Operación de la API `update-network-analyzer-configuration` CLI.

- Creación de una configuración

Cuando crea la configuración, debe especificar un nombre. Por ejemplo, el siguiente comando crea una configuración proporcionando solo un nombre y una descripción opcional. De forma predeterminada, la configuración tiene activada la información de fotogramas y utiliza un nivel de registro de INFO.

```
aws iotwireless create-network-analyzer-configuration \
--configuration-name My_Network_Analyzer_Config \
--description "My first network analyzer configuration"
```

Al ejecutar este comando, se muestra el ARN y el ID de la configuración del analizador de red.

```
{
  "Arn": "arn:aws:iotwireless:us-
east-1:123456789012:NetworkAnalyzerConfiguration/12345678-a1b2-3c45-67d8-e90fa1b2c34d",
  "Id": "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
}
```

- Crear configuración con recursos

Para personalizar los ajustes de configuración, utilice la `trace-content` parámetro. Para añadir recursos, utilice la `wirelessDevices` y `wirelessGateways` parámetros para especificar las puertas de enlace, dispositivos o ambos que desea agregar a la configuración. Por ejemplo, el siguiente comando personaliza los ajustes de configuración y añade a su configuración los recursos inalámbricos especificados por su `WirelessGatewayID` y `WirelessDeviceID`.

```
aws iotwireless create-network-analyzer-configuration \
--configuration-name My_NetworkAnalyzer_Config \
--trace-content WirelessDeviceInfo=DISABLED,LogLevel="ERROR" \
```

```
--wireless-gateways "12345678-a1b2-3c45-67d8-e90fa1b2c34d" "90123456-de1f-2b3b-4c5c-  
bb1112223cd1"  
--wireless-devices "1ffd32c8-8130-4194-96df-622f072a315f"
```

En el ejemplo siguiente se muestra el resultado de la ejecución del comando:

```
{  
    "Arn": "arn:aws:iotwireless:us-  
east-1:123456789012:NetworkAnalyzerConfiguration/12345678-a1b2-3c45-67d8-e90fa1b2c34d",  
    "Id": "12345678-a1b2-3c45-67d8-e90fa1b2c34d"  
}
```

Enumerar las configuraciones del analizador de red

Puede crear varias configuraciones de analizador de red en función de los recursos que deseé supervisar y del nivel de detalle de la información de mensajería de seguimiento que deseé recibir para los recursos. Después de crear estas configuraciones, puede utilizar el [Configuraciones de ListNetworkAnalyzerOperación de la API](#) `list-network-analyzer-configuration-analyzer` Comando de la CLI para obtener una lista de estas configuraciones.

```
aws iotwireless list-network-analyzer-configurations
```

Al ejecutar este comando, se muestran todas las configuraciones del analizador de red de Cuenta de AWS. También puede utilizar `--max-results` para especificar cuántas configuraciones desea mostrar. A continuación, se muestra el resultado de la ejecución de este 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"  
        }  
    ]  
}
```

Eliminar la configuración del analizador de red

Puede eliminar una configuración que ya no utiliza con [Eliminar configuración de NetworkAnalyzerOperación de la API](#) `delete-network-analyzer-configuration` Command de la CLI.

```
aws iotwireless delete-network-analyzer-configuration \  
--configuration-name My_NetworkAnalyzer_Config
```

Si se ejecuta este comando no se generará ningún resultado. Para ver las configuraciones disponibles, puede utilizar la [ListNetworkAnalyzerConfigurationsOperación de la API](#).

Pasos siguientes

Ahora que ha creado una configuración de analizador de red, puede agregar recursos a la configuración o actualizar la configuración. Para obtener más información, consulte [Agregue recursos y actualice la configuración del analizador de red \(p. 1200\)](#).

Agregue recursos y actualice la configuración del analizador de red

Antes de activar la mensajería de seguimiento, agregue los recursos que desea supervisar a la configuración del analizador de red. Los recursos pueden ser uno o ambos dispositivos LoRaWAN y puertas de enlace LoraWAN.

Requisitos previos

Antes de añadir recursos:

1. Debe haber incorporado las puertas de enlace y los dispositivos que desea supervisar en AWS IoT Core de LoRaWAN. Para obtener más información, consulte [Conexión de puertas de enlace y dispositivos a AWS IoT Core for LoRaWAN \(p. 1120\)](#).
2. Debe haber creado una configuración de analizador de red para la que agregará recursos. Para obtener más información, consulte [Creación de una configuración de analizador de red \(p. 1196\)](#).

Agregar recursos y actualizar los ajustes de configuración mediante la consola

Puede agregar recursos y personalizar los parámetros opcionales mediante la AWS IoT consola de AWS IoT API inalámbrica. Además de los recursos, también puede editar los ajustes de configuración y guardar la configuración actualizada.

Agregue recursos a la configuración

1. Abra el icono [Network Analyzer hub](#) de la AWS IoT consola y seleccione la configuración a la que desea agregar recursos.
2. Elija [Actions](#) y luego elija [Add resources](#).
3. Agregue los recursos que desea supervisar mediante la puerta de enlace inalámbrica y los identificadores de dispositivos inalámbricos. Puede elegir varios recursos y agregar hasta 250 puertas de enlace inalámbricas o dispositivos inalámbricos.
4. Una vez añadidos todos los recursos, elija [Add](#).

Verás el número de puertas de enlace y dispositivos que has agregado en la página del hub de Network Analyzer. Puede seguir agregando y eliminando recursos hasta que active la sesión de mensajería de seguimiento. Una vez activada la sesión, para añadir recursos, tendrás que desactivar la sesión.

Actualizar los ajustes de configuración

1. Abra el icono [Network Analyzer hub](#) de la AWS IoT consola y seleccione la configuración para la que desea actualizar los ajustes.
2. Elija [Actions](#) (Acciones) y, a continuación, elija [Edit](#) (Editar).
3. Elija si desea deshabilitar la información de fotogramas y utilizar [Select levels of message detail](#) (Selecione niveles de registro para elegir los niveles de registro que desea utilizar para los registros de mensajes de seguimiento). Seleccione [Save](#) (Guardar).

Verá los valores de configuración especificados en la página de detalles de la configuración del analizador de red.

Agregar recursos y actualizar los ajustes de configuración mediante la API

Para añadir recursos o actualizar los ajustes de configuración, utilice la [Actualización de la configuración de Network Analyzer API](#) o la [actualización-network-analyzer-configuración-analizador CLI](#).

- Actualizar los ajustes de configuración

Para actualizar los ajustes de configuración, utilice la `--trace-content` para especificar el nivel de registro y si se va a habilitar la información de fotogramas. Por ejemplo, el siguiente comando actualiza los valores de configuración desactivando la información de fotogramas y estableciendo el nivel de registro en `ERROR`.

```
aws iotwireless update-network-analyzer-configuration \
--configuration-name NetworkAnalyzerConfig_Default \
--trace-content WirelessDeviceInfo=DISABLED,LogLevel="ERROR"
```

- Añada recursos

Para añadir recursos, utilice la `--wireless-devices-to-add` para especificar las puertas de enlace o dispositivos o ambos que desea agregar a la configuración. Por ejemplo, el siguiente comando actualiza los ajustes de configuración y añade a su configuración los recursos inalámbricos especificados por su `wirelessGatewayID` y `wirelessDeviceID`.

```
aws iotwireless update-network-analyzer-configuration \
--configuration-name NetworkAnalyzerConfig_Default \
--trace-content WirelessDeviceInfo=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 quitar dispositivos o puertas de enlace, utilice el `--wireless-devices-to-remove` y `--wireless-gateways-to-remove` para los parámetros de la API.

Obtener información acerca de la configuración

Ejecutando la `UpdateNetworkAnalyzerConfiguration` API no genera una salida. Para ver los ajustes de configuración y las puertas de enlace o dispositivos que ha agregado, utilice la [Operación de la API get-network-analyzer-configuration](#). Proporcione el nombre de la configuración del analizador de red como entrada.

```
aws iotwireless get-network-analyzer-configuration \
--configuration-name NetworkAnalyzerConfig_Default
```

Si se ejecuta este comando se generará el resultado siguiente.

```
{
  "TraceContent": {
    "WirelessDeviceInfo": "DISABLED",
    "LogLevel": "ERROR"
  },
  "WirelessDevices": [],
  "WirelessGateways": [
    "a0dd70e5-8f15-41a5-89cf-310284e691a1",
    "41682155-de4f-4f8f-84bf-bb5557221fc8"
  ]
}
```

Pasos siguientes

Ahora que ha agregado recursos y ha especificado cualquier configuración opcional para su configuración, puede utilizar el WebSocket protocolo para establecer una conexión con AWS IoT Core para LoRaWAN para utilizar el analizador de redes. A continuación, puede activar la mensajería de seguimiento y empezar

a recibir mensajes de seguimiento de sus recursos. Para obtener más información, consulte [Transmitir mensajes de seguimiento del analizador de red con WebSockets \(p. 1202\)](#).

Transmitir mensajes de seguimiento del analizador de red con WebSockets

Cuando utilice la WebSocket, puede transmitir mensajes de seguimiento del analizador de red en tiempo real. Cuando envía una solicitud, el servicio responde con una estructura JSON. Después de activar la mensajería de seguimiento, puede utilizar los registros de mensajes para obtener información sobre los recursos y solucionar errores. Para obtener más información, consulte [Protocolo WebSocket](#).

A continuación se muestra cómo transmitir mensajes de seguimiento del analizador de red con WebSockets.

Temas

- [Generar una solicitud prefirmada con el WebSocket biblioteca \(p. 1202\)](#)
- [Mensajes y códigos de estado de WebSocket \(p. 1206\)](#)

Generar una solicitud prefirmada con el WebSocket biblioteca

A continuación se muestra cómo generar una solicitud prefirmada para que pueda utilizar el WebSocket biblioteca para enviar solicitudes al servicio.

Añada una política para WebSocket solicitudes para su rol de IAM

Para utilizar el WebSocket protocolo para llamar al analizador de redes, asocie la siguiente política a AWS Identity and Access Management(IAM) que hace esta solicitud.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": "iotwireless:StartNetworkAnalyzerStream",  
            "Resource": "*"  
        }  
    ]  
}
```

Cree una URL prefirmada

Cree una URL para su WebSocket solicitud que contiene la información necesaria para configurar la comunicación entre la aplicación y el analizador de redes. Para verificar la identidad de la solicitud, WebSocket streaming utiliza el proceso de Signature Version 4 de Amazon para firmar solicitudes. Para obtener más información acerca de Signature Version 4, consulte [FirmaAWSSolicitudes API de en la Referencia general de Amazon Web Services](#).

Para llamar al analizador de red, utilice el `StartNetworkAnalyzerStream`URL de la solicitud La solicitud se firmará con las credenciales de la función de IAM mencionada anteriormente. La URL tiene el siguiente formato con saltos de línea añadidos para facilitar la legibilidad. Debe agregar el nombre de la configuración debajo de la línea `&X-Amz-SignedHeaders=host`. Cualquier parámetro adicional debe añadirse debajo de esta línea ordenado por orden alfabético.

En el ejemplo siguiente se muestra cómo utilizar esta URL de solicitud con el nombre de configuración, `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

Si la URL no incluye el nombre de la configuración, AWS IoT Core para LoRaWAN incluirá el nombre predeterminado para la configuración del analizador de red, `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
```

Utilice los valores siguientes para los parámetros de Signature Version 4:

- Algoritmo X-Amz— El algoritmo que estás utilizando en el proceso de firma. El único valor válido es `AWS4-HMAC-SHA256`.
- X-Amz-Credential— Una cadena separada por barras diagonales (`//`) que se forma concatenando los componentes de ID de clave de acceso y ámbito de credenciales. El ámbito de las credenciales incluye la fecha en formato AAAAMMDD, la AWSRegión, nombre del servicio y cadena de terminación (`aws4_request`).
- X-Amz-Date— La fecha y hora en que se creó la firma. Genere la fecha y la hora siguiendo las instrucciones de [Control de fechas en Signature Version 4](#) en la Referencia general de Amazon Web Services.
- X-Amz-Caduca— El tiempo que transcurre en segundos hasta que caduquen las credenciales. El valor máximo es de 300 segundos (5 minutos).
- X-Amz-Security-Token— (opcional) Token de Signature Version 4 para credenciales temporales. Si especifica este parámetro, inclúyalo en la solicitud canónica. Para obtener más información, consulte [Solicitud de credenciales de seguridad temporales](#) en la AWSGuía del usuario de Identity and Access Management.
- X-Amz-Signature— firma de Signature Version 4 que generó para la solicitud.
- Encabezados firmados por X-AMZ— Los encabezados que se firman al crear la firma de la solicitud. El único valor válido es `host`.

Cree la URL de la solicitud y cree firma de Signature Version 4

Para construir la URL de la solicitud y crear la firma de Signature Version 4, utilice los siguientes pasos. Los ejemplos están en pseudocódigo.

Tarea 1: Creación de una solicitud canónica

Cree una cadena que incluya información de su solicitud en un formato estandarizado. Esto garantiza que cuando AWS recibe la solicitud, puede calcular la misma firma que usted calcula en [Tarea 3: Cálculo de la firma \(p. 1205\)](#). Para obtener más información, consulte [Creación de una solicitud canónica para Signature Version 4](#) en la Referencia general de Amazon Web Services.

1. Defina variables para la solicitud en su aplicación.

```
# HTTP verb
method = "GET"
# Service name
service = "iotwireless"
# Región de AWS
region = "Región de 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. Cree un URI canónico (identificador uniforme de recursos). El URI canónico es la parte del URI entre el dominio y la cadena de consulta.

```
canonical_uri = "/start-network-analyzer-stream"
```

3. Cree los encabezados canónicos y los encabezados firmados. Tenga en cuenta la \n final en los encabezados canónicos.

- Añada el nombre de encabezado en minúsculas seguido de un signo de dos puntos.
- Añada una lista de valores separados por comas para ese encabezado. No ordene los valores de los encabezados que tienen múltiples valores.
- Añada una nueva línea (\n).

```
canonical_headers = "host:" + host + "\n"
signed_headers = "host"
```

4. Haga coincidir el algoritmo con el algoritmo de hash. Debe utilizar SHA-256.

```
algorithm = "AWS4-HMAC-SHA256"
```

5. Cree el ámbito de credenciales, que abarca la clave derivada de la fecha, la región y el servicio para el que se realiza la solicitud.

```
credential_scope = datestamp + "/" + region + "/" + service + "/" + "aws4_request"
```

6. Cree la cadena de consulta canónica. Los valores de cadena de consulta deben estar codificados con URI y ordenados por nombre.

- Ordene los nombres de los parámetros en orden ascendente según el punto del código de caracteres. Los parámetros con nombres duplicados deben ordenarse por valor. Por ejemplo, un nombre de parámetro que comienza por la letra mayúscula F precede a un nombre de parámetro que empieza por la letra minúscula b.
- No codifique según las normas de los URI ninguno de los caracteres no reservados que [RFC 3986](#) define: A—Z, a—z, 0—9, guion (-), carácter de subrayado (_), punto (.) y tilde (~).
- Codifique con signos de porcentaje el resto de los caracteres con %XY, donde X e Y son caracteres hexadecimales (0-9 y A-F mayúsculas). Por ejemplo, el carácter de espacio debe codificarse como %20 (no mediante el signo «+» como en algunos esquemas de codificación) y los caracteres extendidos UTF-8 deben indicarse con el formulario %XY%ZA%BC.
- Codifique dos veces los caracteres de equivalencia (=) en los valores de los parámetros.

```
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. Cree un hash de la carga. Para una solicitud GET, la carga es una cadena vacía.

```
payload_hash = HashSHA256("").Encode("utf-8")).HexDigest()
```

8. Combine todos los elementos para crear la solicitud canónica.

```
canonical_request = method + '\n'
    + canonical_uri + '\n'
    + canonical_querystring + '\n'
    + canonical_headers + '\n'
    + signed_headers + '\n'
    + payload_hash
```

Tarea 2: Cree la cadena para firmar

La cadena para firmar contiene metainformación sobre su solicitud. Puede utilizar la cadena para firmar en el siguiente paso cuando calcule la firma de la solicitud. Para obtener más información, consulte [Creación de una cadena para firmar para Signature Version 4](#) en la Referencia general de Amazon Web Services.

```
string_to_sign=algorithm + "\n"
    + amz_date + "\n"
    + credential_scope + "\n"
    + HashSHA256(canonical_request.Encode("utf-8")).HexDigest()
```

Tarea 3: Cálculo de la firma

Genere una clave de firma a partir de la clave de acceso secreta de AWS. Para un mayor grado de protección, la clave derivada es específica de la fecha, el servicio y AWSRegión . Utilice la clave derivada para firmar la solicitud. Para obtener más información, consulte [Cálculo de la firma para AWSSignature Version 4](#) en la Referencia general de Amazon Web Services.

El código se supone que ha implementado la función GetSignatureKey para generar una clave de firma. Para obtener más información y funciones de ejemplo, consulte [Ejemplos de cómo generar una clave de firma para Signature Version 4](#) en la Referencia general de Amazon Web Services.

La función HMAC(key, data) representa una función HMAC-SHA256 que devuelve los resultados en formato binario.

```
#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
```

Tarea 4: Agregar información de firma para solicitar y crear URL de solicitud

Después de calcular la firma, añádasela a la cadena de la solicitud. Para obtener más información, consulte [Agregue la firma a la solicitud](#) en la Referencia general de 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
```

Pasos siguientes

Ahora puede utilizar la URL de solicitud con WebSocket biblioteca para realizar la solicitud al servicio y observar los mensajes. Para obtener más información, consulte [Mensajes y códigos de estado de WebSocket \(p. 1206\)](#).

Mensajes y códigos de estado de WebSocket

Después de crear una solicitud prefirmada, puede utilizar la URL de la solicitud con WebSocket biblioteca o biblioteca que se adapte a su lenguaje de programación, para realizar solicitudes al servicio. Para obtener más información sobre cómo puede generar esta solicitud prefirmada, consulte [Generar una solicitud prefirmada con el WebSocket biblioteca \(p. 1202\)](#).

Mensajes de WebSocket

La WebSocket protocolo se puede utilizar para establecer una conexión bidireccional. Los mensajes se pueden transmitir de cliente a servidor y de servidor a cliente. Sin embargo, el analizador de red solo admite mensajes que se envían de servidor a cliente. Cualquier mensaje recibido del cliente es inesperado y el servidor cerrará automáticamente el WebSocket conexión si se recibe un mensaje del cliente.

Cuando se recibe la solicitud y se inicia una sesión de mensajería de seguimiento, el servidor responde con una estructura JSON, que es la carga útil. Para obtener más información sobre la carga útil y cómo puede activar la mensajería de seguimiento desde el AWS Management Console, consulte [Ver y supervisar los registros de mensajes de seguimiento del analizador de red en tiempo real \(p. 1207\)](#).

Códigos de estado de WebSocket

El ejemplo siguiente muestra la WebSocket códigos de estado para la comunicación del servidor al cliente. La WebSocket Los códigos de estado siguen la[Norma RFC de cierre normal de conexiones](#).

A continuación se muestran los códigos de estado admitidos:

- 1 000

Este código de estado indica un cierre normal, lo que significa que el WebSocket se ha establecido una conexión y se ha cumplido la solicitud. Este estado se puede observar cuando una sesión está inactiva, lo que hace que la conexión se agote el tiempo de espera.

- 1002

Este código de estado indica que el punto final está finalizando la conexión debido a un error de protocolo.

- 1003

Este código de estado indica un estado de error en el que el punto final finalizó la conexión porque recibió datos en un formato que no puede aceptar. El punto final solo admite datos de texto y podría mostrar este código de estado si recibe un mensaje binario o un mensaje del cliente que utiliza un formato no compatible.

- 1008

Este código de estado indica un estado de error en el que el punto final finalizó la conexión porque recibió un mensaje que infringe su política. Este estado es genérico y se muestra cuando los demás

códigos de estado, como 1003 o 1009, no son aplicables. También verás este estado si es necesario ocultar la política o si se produce un error de autorización, como una firma caducada.

- 1011

Este código de estado indica un estado de error en el que el servidor finaliza la conexión porque encontró una condición inesperada o un error interno que le impidió cumplir la solicitud.

Pasos siguientes

Ahora que ha aprendido a generar una solicitud prefirmada y cómo puede observar los mensajes del servidor mediante el WebSocket, puede activar la mensajería de seguimiento y comenzar a recibir registros de mensajes para los recursos de su puerta de enlace inalámbrica y dispositivos inalámbricos. Para obtener más información, consulte [Ver y supervisar los registros de mensajes de seguimiento del analizador de red en tiempo real \(p. 1207\)](#).

Ver y supervisar los registros de mensajes de seguimiento del analizador de red en tiempo real

Si ha agregado recursos a la configuración del analizador de red, puede activar la mensajería de seguimiento para empezar a recibir mensajes de seguimiento de sus recursos. Puede utilizar la AWS Management Console, el AWS IoT API inalámbrica o la AWS CLI.

Requisitos previos

Para poder activar la mensajería de seguimiento mediante el analizador de red, debe tener:

- Se han agregado los recursos que desea supervisar a la configuración predeterminada del analizador de red. Para obtener más información, consulte [Agregue recursos y actualice la configuración del analizador de red \(p. 1200\)](#).
- Se ha generado una solicitud prefirmada mediante el `StartNetworkAnalyzerStreamURL` de la solicitud La solicitud se firmará utilizando las credenciales de la AWS Identity and Access Management tool que hace esta solicitud. Para obtener más información, consulte [Cree una URL prefirmada \(p. 1202\)](#)

Activar la mensajería de seguimiento mediante la consola

Para activar la mensajería de seguimiento

1. Abra el icono **Network Analyzer hub** de la AWS IoT console y elija la configuración del analizador de red, `Network Analyzer Config_default`.
2. En la página de detalles de la configuración del analizador de red, elija **Activar mensajería de seguimiento** y luego elija **Activar**.

Empezarás a recibir mensajes de seguimiento en los que aparece primero el mensaje de seguimiento más reciente en la consola.

Note

Una vez iniciada la sesión de mensajería, la recepción de mensajes de seguimiento puede generar costes adicionales hasta que desactive la sesión o abandone la sesión de seguimiento. Para obtener más información acerca de los precios, consulte [Precios de AWS IoT Core](#).

Ver y supervisar los mensajes de seguimiento

Después de activar la mensajería de seguimiento, el WebSocket se establece la conexión y los mensajes de seguimiento comienzan a aparecer en tiempo real, primero los más recientes. Puede personalizar las preferencias para especificar el número de mensajes de seguimiento que se mostrarán en cada página y para mostrar solo los campos relevantes para cada mensaje. Por ejemplo, puede personalizar el registro de mensajes de seguimiento para mostrar solo los registros de los recursos de puerta de enlace inalámbrica que tienen Log Leve establecido en ERROR, para que pueda identificar y depurar errores rápidamente con sus puertas de enlace. Los mensajes de seguimiento contienen la siguiente información.

- Número de mensaje: Número único que muestra primero el último mensaje recibido.
- ID de recurso: Puerta de enlace inalámbrica o ID de dispositivo inalámbrico del recurso.
- Marca temporal: Hora en que se recibió el mensaje.
- Message ID: Un identificador que AWS IoT Corepara LoRaWAN asigna a cada mensaje recibido.
- Fport: El puerto de frecuencia para comunicarse con el dispositivo mediante el WebSocket conexión.
- DevEui: Identificador único extendido (EUI) de su dispositivo inalámbrico.
- Recurso: Si el recurso supervisado es un dispositivo inalámbrico o una puerta de enlace inalámbrica.
- Evento: El evento de un mensaje de registro de un dispositivo inalámbrico, que puede ser Unirse, Volver a unirse, Uplink_Data, Downlink_Data, o bien Registration (Registro).
- Log Level: Información sobre INFO, ERROR o SECUNDARIAS de registro para su dispositivo.

Mensaje de registro JSON del analizador de red

También puede elegir un mensaje de seguimiento a la vez para ver la carga útil JSON de ese mensaje. Según el mensaje que seleccione en los registros de mensajes de seguimiento, verá información en la carga útil JSON que indica que contiene 2 partes: Registro de clientes y Marco Lora.

Registro de clientes

La Registro de clientes parte del JSON muestra el tipo y el identificador del recurso que recibió el mensaje, el nivel de registro y el contenido del mensaje. En el siguiente ejemplo, se muestra un Registro de clientes Mensaje de registro. Puede utilizar el message en el JSON para obtener más información sobre el error y cómo se puede resolver.

Marco Lora

La Marco Lora parte del JSON tiene un Message ID y contiene información sobre la carga útil física del dispositivo y los metadatos inalámbricos.

A continuación, se muestra la estructura del mensaje de seguimiento.

```
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;
},
};
```

Revisión y siguientes pasos

En esta sección, ha visto los mensajes de seguimiento y ha aprendido cómo utilizar la información para depurar errores. Una vez que hayas visto todos los mensajes, puedes:

- Desactivar la mensajería de seguimiento

Para evitar costes adicionales, puede desactivar la sesión de mensajería de seguimiento. Al desactivar la sesión, se desconecta el WebSocket conexión para que no recibas ningún mensaje de seguimiento adicional. Puede seguir viendo los mensajes existentes en la consola.

- Editar información de fotogramas para la configuración

Puede editar la configuración del analizador de red y elegir si desea desactivar la información de fotogramas y elegir los niveles de registro de los mensajes. Antes de actualizar la configuración, considere la posibilidad de desactivar la sesión de mensajería de seguimiento. Para realizar estas ediciones, abra el[Página de detalles de Network Analyzer en elAWS IoTconsolay eligeEditar](#). A continuación, puede actualizar la configuración con los nuevos ajustes de configuración y activar la mensajería de seguimiento para ver los mensajes actualizados.

- Agregue recursos a la configuración

También puede agregar más recursos a la configuración del analizador de red y supervisarlos en tiempo real. Puede agregar hasta un total combinado de 250 recursos de puerta de enlace inalámbrica y dispositivos inalámbricos. Para añadir recursos, en la[Página de detalles de Network Analyzer deAWS IoTconsola](#), elige elRecursosPestaña y elijaAñada recursos. A continuación, puede actualizar la configuración con los nuevos recursos y activar la mensajería de seguimiento para ver los mensajes actualizados de los recursos adicionales.

Para obtener más información sobre cómo actualizar la configuración del analizador de red mediante la edición de los ajustes de configuración y la adición de recursos, consulte[Agregue recursos y actualice la configuración del analizador de red \(p. 1200\)](#).

Seguridad de los datos conAWS IoT Corefor LoRaWAN

Dos métodos protegen los datos de suAWS IoT Corefor LoRaWAN:

- La seguridad que utilizan los dispositivos inalámbricos para comunicarse con las puertas de enlace.

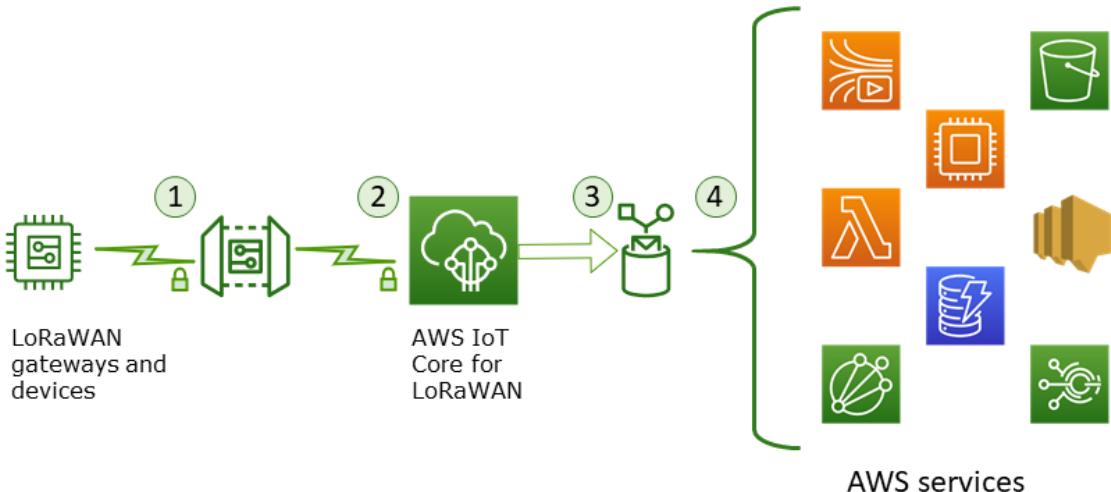
Los dispositivos LoraWAN siguen las prácticas de seguridad descritas en[SEGURIDAD LoRaWAN™: Libro blanco preparado para el LoRa Alliance™ de Gemalto, Actility y Semtech](#)para comunicarse con las puertas de enlace.

- La seguridad que AWS IoT Core utiliza para conectar puertas de enlace a AWS IoT Core para LoRaWAN y envíe los datos a otros AWS Servicios de .

AWS IoT Core seguridad se describe en [Protección de los datos en AWS IoT Core \(p. 381\)](#).

Cómo se protegen los datos en todo el sistema

Este diagrama identifica los elementos clave de un sistema LoRaWAN conectado a AWS IoT Core para que LoRaWAN identifique cómo se protegen los datos en todo momento.



1. El dispositivo inalámbrico LoRaWAN cifra sus mensajes binarios utilizando el modo CTR AES128 antes de transmitirlos.
2. Conexiones de gateway a AWS IoT Core para LoRaWAN están protegidos por TLS como se describe en [Seguridad de transporte en AWS IoT \(p. 382\)](#). AWS IoT Core para LoRaWAN descifra el mensaje binario y codifica la carga útil del mensaje binario descifrada como una cadena base64.
3. El mensaje codificado en base64 resultante se envía como carga útil del mensaje a AWS IoT Core para LoRaWAN, que lo encierra en una estructura de mensaje JSON y lo manda a AWS IoT Core para LoRaWAN. Los datos dentro de AWS IoT Core para LoRaWAN están encriptados mediante AWS IoT Core para LoRaWAN llaves de propiedad.
4. AWS IoT Core para LoRaWAN dirige los datos del mensaje a los servicios descritos en la configuración de la regla. Los datos dentro de AWS IoT Core para LoRaWAN están encriptados mediante AWS IoT Core para LoRaWAN llaves de propiedad.

Seguridad del transporte de dispositivos y pasarela LoRaWAN

Dispositivos LoRaWAN y AWS IoT Core para LoRaWAN almacena claves raíz previamente compartidas. Las claves de sesión se derivan de los dispositivos LoRaWAN y AWS IoT Core para LoRaWAN siguiendo los protocolos. Las claves de sesión simétricas se utilizan para cifrar y descifrar en un modo CTR estándar AES-128. También se utiliza un código de integridad de mensajes (MIC) de 4 bytes para comprobar la integridad de los datos siguiendo un algoritmo CMAC estándar AES-128. Las claves de sesión se pueden actualizar mediante el proceso Unir/Reunirse.

La práctica de seguridad para LoRa las puertas de enlace se describen en las especificaciones de LoRaWAN. LoRa Gateways se conectan a AWS IoT Core para LoRaWAN a través de un socket web utilizando un [AWS IoT Core for LoRaWAN](#). AWS IoT Core para LoRaWAN admite únicamente [AWS IoT Core for LoRaWAN](#) versión 2.0.4 y versiones posteriores.

Antes de establecer la conexión de socket web,AWS IoT Corefor LoRaWAN utiliza[Modo de autenticación de cliente y servidor TLS \(p. 382\)](#)para autenticar la puerta de enlace.AWS IoT Corepara LoRaWAN también mantiene un servidor de configuración y actualización (CUPS) que configura y actualiza los certificados y claves utilizados para la autenticación TLS.

Integración de Amazon Sidewalk paraAWS IoT Core

Amazon Sidewalk es una red compartida que ayuda a dispositivos como Amazon Echo, cámaras de seguridad Ring, luces exteriores, sensores de movimiento y rastreadores de baldosas a funcionar mejor en casa y fuera de la puerta principal. Cuando está habilitada, esta red puede admitir otros dispositivos Sidewalk de su comunidad y abrir la puerta a innovaciones, como la localización de artículos conectados a Amazon Sidewalk. Amazon Sidewalk ayuda a que sus dispositivos se conecten y permanezcan conectados. Por ejemplo, si el dispositivo pierde su conexión Wi-Fi, Sidewalk puede simplificar la reconexión al router. Para obtener más información, consulte [Guía de inicio rápido de Amazon Sidewalk](#).

En las secciones siguientes se muestra cómo incorporar los dispositivos Sidewalk con AWS IoT y usar notificaciones de eventos para notificarte de eventos, como cuando el dispositivo Sidewalk está registrado. Para obtener información sobre el uso de Amazon CloudWatch para supervisar sus dispositivos Sidewalk, consulte [Monitoreo y registro de AWS IoT Wireless usando Amazon CloudWatch \(p. 1220\)](#).

Cómo incorporar tu dispositivo Sidewalk

Puedes incorporar tus dispositivos Sidewalk en AWS IoT utilizando la consola de AWS IoT o la AWS IoT API inalámbrica. Después de autenticar los dispositivos Amazon Sidewalk, sus mensajes se envían a AWS IoT Core. A continuación, puedes comenzar a desarrollar tus aplicaciones empresariales en el AWS Cloud, que utilizan los datos de tus dispositivos Amazon Sidewalk.

Con la consola

Para incorporar los dispositivos Sidewalk mediante la AWS Management Console, primero registre su dispositivo en la [Consola de Sidewalk Developer Service \(SDS\)](#) y, a continuación, asocie su ID de Amazon a su cuenta de AWS. Para ver los dispositivos Sidewalk que has agregado y gestionarlos, inicia sesión en la AWS Management Console y vaya a la [Dispositivos](#) en la AWS IoT consola de .

Uso de la API o la CLI

Puedes incorporar dispositivos Sidewalk y LoRaWAN mediante la [AWS IoTWireless API](#). La AWS IoT API inalámbrica que AWS IoT Core soporta es la que se basa en el AWS SDK para el AWS IoT. Para obtener más información, consulta [AWS SDK y conjuntos de herramientas](#).

Puedes utilizar la AWS CLI para ejecutar comandos de incorporación y administración de dispositivos Sidewalk. Para obtener más información, consulta [AWS IoT Reference de la CLI inalámbrica](#).

Dispositivos Sidewalk integrados con integración de Amazon Sidewalk paraAWS IoT Core

Con la integración de Amazon Sidewalk para AWS IoT Core, puedes agregar tu flota de dispositivos Sidewalk a la AWS Cloud. Para comenzar a utilizar, sigue estos pasos.

1. Revisar el SDK de Sidewalk y la documentación

Más información sobre Amazon Sidewalk y cómo pueden utilizarlo sus dispositivos.

- a. Consulte la Guía de inicio rápido de Amazon Sidewalk.
 - b. Descargue un SDK para Amazon Sidewalk.
 - c. Abra el icono[Consola de Sidewalk Developer Service \(SDS\)](#).
2. Registrar su prototipo de dispositivo

En la consola de SDS, registre su dispositivo prototipo con Amazon Sidewalk.
 3. Asocia tu ID de Amazon de Sidewalk con tuCuenta de AWS

En el navegador[AWS IoTconsola](#), asocia tu ID de Amazon de Sidewalk con tuCuenta de AWS.

Los dispositivos Amazon Sidewalk aparecen en elAceraPestaña de laDispositivosCentro de la[AWS IoTconsola](#).
 4. Complete la configuración del dispositivo Amazon Sidewalk en el[AWS IoTconsola](#)

Cree los destinos y reglas que necesita su dispositivo Sidewalk para enrutar y dar formato a los datos paraAWServicios de .

En los temas siguientes se muestra cómo agregar dispositivos Sidewalk y conectarlos aAWS IoT. Antes de añadir los dispositivos, asegúrese de que suCuenta de AWStiene los permisos de IAM requeridos para realizar lo siguiente:procedimientos.

Temas

- [Agregue las credenciales de su cuenta de Sidewalk \(p. 1213\)](#)
- [Añadir un destino para su dispositivo Sidewalk \(p. 1214\)](#)
- [Crear reglas para procesar mensajes de dispositivos Sidewalk \(p. 1216\)](#)

Agregue las credenciales de su cuenta de Sidewalk

Puede conectar dispositivos Sidewalk aAWS IoTmediante el uso de laAWS Management Consoleo elAWS IoTAPI inalámbrica. Para incorporar el dispositivo, crearemos un perfil de conectividad inalámbrica para su dispositivo Sidewalk y, a continuación, agregaremos un destino yAWS IoTregla para el perfil y los extremos de acera.

Para añadir el dispositivo, debe añadir las credenciales de su cuenta de Sidewalk. Puede añadir las credenciales utilizando laAWS Management Consoleo elAWS IoTAPI inalámbrica.

Agregue las credenciales de su cuenta de Sidewalk utilizando la consola

Para agregar las credenciales de su cuenta de Sidewalk desde la consola:

1. Vaya a la [.Perfiles](#)Página de laAWS IoTConsola de y elija laAceraPestaña.

Note

Asegúrese de utilizar laus-east-1Región . Esta pestaña no aparece en la consola si utilizas una región diferente.

2. Introduce el ID de Amazon de Sidewalk. Obtienes esta identificación del[Consola de Sidewalk Developer Service \(SDS\)](#)al diseñar su producto Sidewalk. Para obtener más información, consulte[Diseñe su producto Sidewalk](#).

3. Cargar laClave privada de AppServer, que es la clave de servidor que proporcionó su proveedor. LaClave privada de AppServeres la clave privada ED25519 (laapp-server-ed25519-private.txtfile), que es un valor hexadecimal de 64 dígitos. Ha generado esta clave utilizando la herramienta de generación de certificados Sidewalk cuando diseñó su producto Sidewalk. Para obtener más información, consulte [Diseñe su producto Sidewalk](#).
4. Para añadir sus credenciales de Sidewalk, elijaAdición de credencial.

Agregue las credenciales de su cuenta de Sidewalk utilizando la API

Puede utilizar elAWS IoTAPI inalámbrica para agregar las credenciales de su cuenta de Sidewalk. En la siguiente lista se describen las acciones de la API.

AWS IoTAcciones de API inalámbrica para la cuenta Sidewalk

- [Asociar una cuenta AWS con una cuenta de socio](#)
- [Desasociar una cuenta AWS de la cuenta de socio](#)
- [Obtener cuenta de socio](#)
- [Listar cuentas de socios](#)
- [Actualizar cuenta de socio](#)

Para ver la lista completa de las acciones y los tipos de datos disponibles para crear y administrarAWS IoT CorePara obtener información sobre los recursos de LoRaWAN, consulte la[AWS IoTReferencia de la API inalámbrica](#).

Describe cómo utilizar laAWS CLIpara añadir una cuenta

Puede utilizar elAWS CLIpaa asociar una cuenta de Sidewalk a suCuenta de AWSmediante el uso de [laasociar-aws-cuenta-con-cuenta de socio](#), como se ilustra en el siguiente ejemplo.

Note

También puede realizar este procedimiento con la API utilizando los métodos de la API de AWS que corresponden a los comandos CLI que se muestran aquí.

```
aws iotwireless associate-aws-account-with-partner-account \
  --sidewalk
  AmazonId="12345678901234",AppServerPrivateKey="a123b45c6d78e9f012a34cd5e6a7890b12c3d45e6f78a1b234c56d7
```

Pasos siguientes

Ahora que ha agregado las credenciales y ha configurado un perfil de conectividad inalámbrica, puede agregar un destino para su dispositivo Sidewalk. Verás las credenciales que has agregado en el[PerfilesPágina de laAWS IoTConsola de](#), en laAceraPestaña. Definirá un nombre de rol y un nombre de regla para el destino que puede enrutar los mensajes enviados desde sus dispositivos. Para obtener más información, consulte [Añadir un destino para su dispositivo Sidewalk \(p. 1214\)](#).

Añadir un destino para su dispositivo Sidewalk

Antes de poder añadir unAWS IoT Corepara el destino LoraWAN y cree una regla para enrutar los mensajes enviados desde su dispositivo Sidewalk, debe crear un perfil de conectividad inalámbrica.

Para crear el perfil, primero registre el dispositivo Sidewalk y, a continuación, agregue las credenciales a su Cuenta de AWS. Para obtener más información, consulte [Agregue las credenciales de su cuenta de Sidewalk \(p. 1213\)](#).

La creación de un destino de acera es similar a la forma en que crea un destino para sus dispositivos LoRaWan. A continuación se muestra cómo crear un destino utilizando la AWS Management Console o la API.

Agregue un destino utilizando la consola

Puede agregar su destino de acera desde la [Página Destinos](#) de la AWS IoT consola de .

Especifique los siguientes campos al crear un AWS IoT Core para el destino LoRaWan y, a continuación, elija Adición de destino.

- Detalles del destino

Introduzca un Nombre de destino y una descripción opcional para su destino. Para el registro Nombre de destino, introduzca **SidewalkDestination**. De forma opcional, puede especificar una descripción, como **This is a destination for Sidewalk devices**.

- Nombre de la regla

La AWS IoT regla configurada para procesar los datos del dispositivo. El destino necesita una regla para procesar los mensajes que recibe. Introduzca el nombre de una regla (digamos **SidewalkRule**) y luego elija Copiar para copiar el nombre de la regla que introducirás al crear la AWS IoT regla. Puede elegir Crear regla para crear la regla ahora o navegar hasta la [Reglas](#) Centro de la AWS IoT consola y cree una regla con el nombre que ha copiado.

Para obtener más información acerca de las reglas para destinos, consulte [Crear reglas para procesar mensajes de dispositivos Sidewalk](#).

- Role name (Nombre de rol)

El rol de IAM que otorga permiso a los datos del dispositivo para tener acceso a la regla mencionada en Nombre de la regla. Para crear el rol de IAM, siga los pasos descritos en [Crear un rol de IAM para sus destinos \(p. 1136\)](#). Al crear el rol:

- Para seleccionar el tipo de entidad de confianza, elige Servicio de AWS, a continuación, elija IoT como servicio.
- Entrar **SidewalkRole** para la Nombre del rol.
- Utilice el mismo documento de política que se describe en [Crear un rol de IAM para sus destinos \(p. 1136\)](#).

Para obtener más información acerca de los roles de IAM, consulte [Uso de roles de IAM](#).

Agregar un destino mediante la API

En las listas siguientes se describen las acciones de la API que realizan las tareas asociadas con la adición, actualización o eliminación de un destino.

AWS IoT Acciones de API inalámbrica para perfiles de servicio

- [Crear destino](#)
- [Obtener destino](#)
- [Listar destinos](#)
- [UpdateDestination](#)

- [DeleteDestination](#)

Para ver la lista completa de las acciones y los tipos de datos disponibles para crear y administrar AWS IoT CorePara obtener información sobre los recursos de LoRaWAN, consulte la[AWS IoT Referencia de la API inalámbrica](#).

Describe cómo utilizar laAWS CLIpaañadir un destino

Puede utilizar elAWS CLIpaañadir un destino mediante el[crear destino](#) comando. En el ejemplo siguiente se crea un destino.

```
aws iotwireless create-destination \  
  --name SidewalkDestination \  
  --expression-type RuleName \  
  --expression SidewalkRule \  
  --role-arn arn:aws:iam::123456789012:role/SidewalkRole
```

Al ejecutar este comando se crea un destino con el nombre de destino, el nombre de la regla y el nombre de la función especificados. Para obtener información sobre los nombres de reglas y roles de los destinos, consulte[Crear reglas para procesar mensajes de dispositivos Sidewalk](#).

Para obtener más información sobre los CLI que puede utilizar, consulte[AWS CLI Referencia de de](#).

Pasos siguientes

Ahora que ha agregado el destino, puede crear la regla de destino para su dispositivo Sidewalk que enrutará los mensajes a otros servicios. Para obtener más información, consulte [Crear reglas para procesar mensajes de dispositivos Sidewalk \(p. 1216\)](#).

Crear reglas para procesar mensajes de dispositivos Sidewalk

AWS IoTLas reglas pueden recibir los mensajes de los dispositivos Sidewalk y enrutarlos a otros servicios.[AWS IoT Corepara destinos LoRaWAN \(p. 1134\)](#)asociar un dispositivo Sidewalk con la regla que procesa los datos de mensajes del dispositivo para enviarlos a otros servicios.

Puede utilizar una regla existente para su destino. En esta sección, vamos a crear la regla,SidewalkRule, que especificó al crear el destino de acera, tal y como se describe en[Añadir un destino para su dispositivo Sidewalk \(p. 1214\)](#). Al crear la regla, crearemos unAWS Lambda para volver a publicar el mensaje en unAWS IoT tema.

Crear una regla de destino de acera

Vaya a la [.Reglas](#)Centro de laAWS IoTrealice los siguientes pasos.

1. ElegirCreación de una reglapara crear una regla nueva para el destino.
2. Introduzca el nombre**SidewalkRule**para laNombrey especifique un opcionalDescripciónpara la regla (por ejemplo,**Sidewalk rule for lambda action to republish a topic**).
3. Cambiar la instrucción de consulta predeterminada a**SELECT ***de modo que se realice cualquier acción asociada a la regla. Mantenga la versión SQL en2016-03-23.
4. En Set one or more actions (Definir una o varias acciones), elija Add action (Añadir acción).
5. Para la acción de regla, seleccioneEnviar un mensaje a una función Lambda y luego elijaConfiguración de una acción.

6. Puede elegir una función Lambda existente o crear una nueva. En este ejemplo, vamos a crear una función Lambda. ElegirCrear una nueva función de Lambda.

Cree su función usandoAWS Lambda

ElegirCrear una nueva función de Lambdaabre la[Funciones](#)de la consola de Lambda. Siga estos pasos.

1. Para crear su función, elijaAuthor from scratch.
2. ParaNombre de la función, escriba un nombre (por ejemplo,**Sidewalk_Handler**), eligePython 3 . 8comoRuntime (Tiempo de ejecución):y, a continuación, elijaCreación de una función.
3. Elija el iconolambda.py que esfunción en laFuente de código sección de la consola de.
4. En el cuerpo de la función, elimine cualquier código dentro del cuerpo de la función y agregue una instrucción de impresión para la función Lambda. También puedes usar base64 para decodificar elPayloadDatapara recibir los datos de la aplicación a los que envía el dispositivoAWS IoT. A continuación se muestra un ejemplo de la función 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 desplegar el código de función, elijadesplegar.
6. Vuelva a la[ReglasHub](#) de la consola y actualice la página. Elija la función Lambda que ha creado y elijaAdición de acción.

Vuelva a publicar un mensaje en unAWS IoTTema de

Puede añadir una segunda acción para volver a publicar un mensaje en unAWS IoTTema del[ReglasHub](#) de la consola.

1. Elija Añadir acción.
2. ElegirVuelva a publicar un mensaje en unAWS IoTTema dey eligeConfiguración de una acción.
3. Entrar[project/sensor/observed](#)para laTemay asegúrese de que laCalidad del serviciotoma el valor0 - El mensaje se entrega cero o más veces.
4. Elija Create Role (Crear rol). EntrarFunción de publicación de Sidewalkpara el nombre del rol y elijaCreación de un rol.
5. Elija Añadir acción.

Ambas acciones aparecen en elReglasCentro de laAWS IoTconsola de .

6. Elija Create rule (Crear regla).

La regla aparece en elReglasen la que se muestra la lista de reglas.

Pasos siguientes

Ahora que ha creado la regla de destino para su dispositivo Sidewalk, puede conectar el dispositivo y observar los mensajes sobre el tema al que se suscribió. Para obtener más información, consulte [Connect tu dispositivo Sidewalk y visualiza el formato de metadatos de enlace ascendente \(p. 1218\)](#).

Connect tu dispositivo Sidewalk y visualiza el formato de metadatos de enlace ascendente

Después de agregar las credenciales de Sidewalk y agregar el destino, puede aprovisionar los puntos finales de Sidewalk y conectar el dispositivo.

Connect el dispositivo Sidewalk

Puede aprovisionar su dispositivo como endpoint Sidewalk generando los certificados de dispositivo y los certificados de servidor de aplicaciones desde el[Consola de Sidewalk Developer Service \(SDS\)](#). Para obtener más información, consulte[Aprovisionamiento y configuración de los endpoints de acera](#).

Después de conectar el dispositivo, verá su dispositivo Sidewalk en la[Dispositivos](#)Página de laAWS IoTConsola de, en laAceraPestaña. Cuando tu dispositivo esté conectado y empiece a enviar datos, verás la fecha y la hora delÚltimo enlace ascendente recibido en.

Ver formato de los mensajes de enlace ascendente

Una vez conectado el dispositivo, puede suscribirse al tema (por ejemplo,`project/sensor/observed`) que especificó al crear la regla de destino de acera y observe los mensajes de enlace ascendente del dispositivo. Para suscribirse al tema, vaya a la[Cliente de prueba MQTT](#)en elPruebasPágina de laAWS IoT, escriba el nombre del tema (por ejemplo,`project/sensor/observed`) y, a continuación, elijaSuscribirse.

En el ejemplo siguiente se muestra el formato de los mensajes de enlace superior que se envían desde los dispositivos Sidewalk aAWS IoT. LaWirelessMetadatacontiene los metadatos sobre la solicitud de mensajes.

```
{  
    "PayloadData": "ZjRlNjY1ZWNlNw==",  
    "WirelessDeviceId": "6dc562bf-31c5-37e0-b230-716ea8148c3d",  
    "TransmitMode": "0",  
    "WirelessMetadata": {  
        "Sidewalk": {  
            "CmdExStatus": "Cmd",  
            "SidewalkId": "device-id",  
            "Seq": 0,  
            "MessageType": "messageType"  
        }  
    }  
}
```

En la tabla siguiente se muestra una definición de los distintos parámetros de los metadatos del vínculo superior. La`device-id`s el ID del dispositivo inalámbrico, como`ABCDEF1234`y la`messageType`es el tipo de mensaje de enlace superior que se recibe del dispositivo.

Parámetros de metadatos de enlace ascendente de acera

Parámetro	Descripción	Tipo	Obligatorio
<code>PayloadData</code>	Carga útil de mensajes que se envía desde el dispositivo inalámbrico.	Cadena	Sí
<code>WirelessDeviceID</code>	Identificador del dispositivo inalámbrico que envía los datos	Cadena	Sí

Parámetro	Descripción	Tipo	Obligatorio
TransmitMode	Modo de transmisión de los datos que se envían desde el dispositivo inalámbrico. Puede ser 0 para un confirmed modo, 1 para confirmed, y 2 para un unused.	Entero	Sí
Sidewalk.CmdExStatus	Estado del tiempo de ejecución del comando. Los mensajes de tipo respuesta incluirán el código de estado, COMMAND_EXEC_STATUS_SUCCESS. Sin embargo, es posible que las notificaciones no incluyan el código de estado.	Enumeración	No
Sidewalk.NackExStatus	Estado de respuesta, que puede ser RADIO_TX_ERROR o MEMORY_ERROR.	Matriz de cadenas	No

Monitoreo y registro de AWS IoT Wireless con Amazon CloudWatch

Puede monitorizar su AWS IoT Wireless recursos y aplicaciones que se ejecutan en tiempo real mediante Amazon CloudWatch. Puede monitorear el estado de sus dispositivos LoraWan y Sidewalk que ha incorporado con Amazon CloudWatch.

- Para obtener información sobre la incorporación de dispositivos LoraWAN, consulte [Conexión de puertas de enlace y dispositivos a AWS IoT Core for LoRaWAN \(p. 1120\)](#).
- Para obtener información sobre la incorporación de dispositivos Amazon Sidewalk en AWS IoT Core, consulte [Dispositivos Sidewalk integrados con integración de Amazon Sidewalk para AWS IoT Core \(p. 1212\)](#).

Usar CloudWatch para recopilar y hacer un seguimiento de métricas, que son las variables que puede medir en sus recursos y aplicaciones. Para obtener más información sobre las ventajas de utilizar la supervisión, consulte [Monitorización de AWS IoT \(p. 426\)](#).

Si desea obtener más información de registro en tiempo real de sus dispositivos LoraWAN, utilice el analizador de red. Para obtener más información, consulte [Supervisión de su flota de recursos inalámbricos en tiempo real mediante el analizador de redes \(p. 1193\)](#).

Cómo monitorizar los recursos inalámbricos

Para registrar y monitorear sus recursos inalámbricos, siga estos pasos.

1. Cree un rol de registro para registrar su AWS IoT Wireless recursos, tal como se describe en [Creación de una política y un rol de registro para AWS IoT Wireless \(p. 1221\)](#).
2. Mensajes de registro en CloudWatch La consola de registros tiene un nivel de registro predeterminado de **ERROR**, que es menos detallado y solo contiene información de error. Si desea ver mensajes más detallados, le recomendamos que utilice la CLI para configurar el registro primero, tal y como se describe en [Configuración de registro en AWS IoT Wireless recursos \(p. 1223\)](#).
3. A continuación, puede supervisar sus recursos viendo las entradas de registro en el CloudWatch Consola de registros. Para obtener más información, consulte [Vista CloudWatch AWS IoT Wireless Entradas de registro de en \(p. 1232\)](#).
4. Puede crear expresiones de filtro mediante Grupos de registros pero le recomendamos que primero cree filtros simples y vea las entradas de registro en los grupos de registros y, a continuación, vaya a CloudWatch Perspectivas para crear consultas para filtrar las entradas de registro en función del recurso o evento que esté supervisando. Para obtener más información, consulte [Usar CloudWatch Perspectivas para filtrar registros para AWS IoT Wireless \(p. 1238\)](#).

Los siguientes temas muestran cómo configurar el registro de AWS IoT Wireless para recopilar métricas de CloudWatch. Además de los dispositivos LoraWAN, puede utilizar estos temas para configurar el registro de cualquier dispositivo Sidewalk que haya agregado a su cuenta y supervisarlos. Para obtener información sobre cómo agregar estos dispositivos, consulte [Integración de Amazon Sidewalk para AWS IoT Core \(p. 1212\)](#).

Temas

- [Configuración del registro para AWS IoT Wireless \(p. 1221\)](#)

- MonitorAWS IoT Wireless con CloudWatch Registros (p. 1231)

Configuración del registro para AWS IoT Wireless

Antes de poder supervisar y registrar AWS IoT actividad, activar primero el registro para AWS IoT Wireless recursos mediante la CLI o la API.

Al considerar cómo configurar su AWS IoT Wireless registro, la configuración de registro predeterminada determina cómo AWS IoT La actividad se registrará a menos que especifique lo contrario. Para empezar, puede que desee obtener registros detallados con un nivel de registro predeterminado de `INFO`.

Después de revisar los registros iniciales, puede cambiar el nivel de registro predeterminado a `ERROR`, que es menos detallado y establece un nivel de registro específico de recursos más detallado en los recursos que puedan necesitar más atención. Los niveles de registro se pueden cambiar cuando lo deseé.

Los siguientes temas muestran cómo configurar el registro de AWS IoT Wireless de AWS.

Temas

- Creación de una política y un rol de registro para AWS IoT Wireless (p. 1221)
- Configuración de registro en AWS IoT Wireless recursos (p. 1223)

Creación de una política y un rol de registro para AWS IoT Wireless

A continuación se muestra cómo crear un rol de registro solo AWS IoT Wireless de AWS. Si desea crear también un rol de registro para AWS IoT Core, consulte [Creación de un rol de registro \(p. 427\)](#).

Creación de un rol de registro para AWS IoT Wireless

Antes de habilitar el registro, debe crear un rol de IAM y una política de que otorgue AWS permiso para supervisar AWS IoT Wireless actividad en su nombre.

Creación del rol de IAM para registrar

Para crear un rol de registro para AWS IoT Wireless, abra el [Centro de roles de la consola de IAM](#) y elige Creación de un rol.

1. Under Seleccione el tipo de entidad de confianza, elige Otro AWS cuenta.
2. En ID de cuenta, introduzca su AWS ID de cuenta y, a continuación, elija Siguiente: Permisos.
3. En el cuadro de búsqueda, escriba **AWSIoTWirelessLogging**.
4. Seleccione el cuadro situado junto a la política denominada `AWSIoTWirelessLogging` y luego seleccione Siguiente: Tags (Etiquetas).
5. Seleccione Next (Siguiente): Consulte.
6. En Nombre del rol, introduzca **IoTWirelessLogsRole** y luego seleccione Creación de un rol.

Modificar la relación de confianza del rol de IAM

En el mensaje de confirmación que aparece después de ejecutar el paso anterior, elija el nombre del rol que creó, Función de registros inalámbricos de IoT. A continuación, editará el rol para añadir la siguiente relación de confianza.

1. En el navegadorResumensección del rolFunción de registros inalámbricos de IoT, elige elRelaciones de confianzay, a continuación, seleccioneModificar relación de confianza.
2. EnPolicy Document, cambie elPrincipalpropiedad que tenga el aspecto de este ejemplo.

```
"Principal": {  
    "Service": "iotwireless.amazonaws.com"  
},
```

Después de cambiar elPrincipal, el documento de política completo debe tener el aspecto del siguiente ejemplo.

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

3. Para guardar los cambios y salir de, elijaActualización de la política de confianza.

Política de registro paraAWS IoT Wireless

En el siguiente documento de política se proporciona la política de rol y de confianza que permiteAWS IoT Wirelessenviar entradas de registro a CloudWatch en su nombre.

Note

EsteAWSEl documento de política administrado se creó automáticamente para usted cuando creó el rol de registro,Función de registros inalámbricos de IoT.

Política de roles de

A continuación se muestra el documento de política de roles.

```
{  
    "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 confianza para iniciar sesión únicamenteAWS IoT Wirelessactividad

A continuación se muestra la política de confianza solo para el registroAWS IoT Wirelessactividad.

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

Si ha creado el rol de IAM para registrar tambiénAWS IoT Corey, a continuación, los documentos de política le permiten registrar ambas actividades. Para obtener información sobre la creación de un rol de registro paraAWS IoT Core, consulte[Creación de un rol de registro \(p. 427\)](#).

Pasos siguientes

Ha aprendido a crear un rol de registro para registrar suAWS IoT Wirelessde AWS. De forma predeterminada, los registros tienen un nivel de registro de**ERROR**, así que si quieras ver solo la información de errores, ve a[Vista CloudWatch AWS IoT WirelessEntradas de registro de en \(p. 1232\)](#)para supervisar los recursos inalámbricos mediante la visualización de las entradas de registro.

Si desea obtener más información en las entradas de registro, puede configurar el nivel de registro predeterminado para sus recursos o para distintos tipos de eventos, como establecer el nivel de registro en**INFO**. Para obtener información sobre cómo configurar el registro de sus recursos, consulte[Configuración de registro enAWS IoT Wirelessrecursos \(p. 1223\)](#).

Configuración de registro enAWS IoT Wirelessrecursos

Para configurar el registro deAWS IoT Wirelessrecursos, puede utilizar la API o la CLI. Al empezar a monitorizarAWS IoT Wirelessrecursos, puede utilizar la configuración predeterminada. Para ello, puede saltarse este tema y proceder a[MonitorAWS IoT Wirelesscon CloudWatch Registros \(p. 1231\)](#)para supervisar los registros.

Después de comenzar a supervisar los registros, puede utilizar la CLI para cambiar los niveles de registro a una opción más detallada, como proporcionar**INFO** y **ERROR** información y habilitación del registro para obtener más recursos.

AWS IoT Wirelessrecursos y niveles de registros

Antes de utilizar la API o la CLI, utilice la siguiente tabla para obtener información sobre los distintos niveles de registro y los recursos para los que puede configurar el registro. En la tabla se muestran los parámetros que aparecen en el CloudWatch registra cuando monitoriza los recursos. La forma en que configura el registro de los recursos determinará los registros que ve en la consola.

Para obtener información sobre qué muestra CloudWatch el aspecto de los logs y cómo puede utilizar estos parámetros para registrar información útil sobre elAWS IoT Wirelessrecursos, consulte[Vista CloudWatch AWS IoT WirelessEntradas de registro de en \(p. 1232\)](#).

Niveles de registro y recursos

Nombre	Valores posibles	Descripción
logLevel	INFO, ERROR o DISABLED	<ul style="list-style-type: none"> ERROR: muestra cualquier error que provoque el fracaso de una operación. Los registros solo incluyen ERROR información. INFO: proporciona información de alto nivel sobre el flujo de objetos. Los registros incluyen INFO y ERROR información. DISABLED: Desactiva todos los registros.
resource	WirelessGateway o WirelessDevice	El tipo de recurso, que puede ser <code>WirelessGateway</code> o <code>WirelessDevice</code> .
wirelessGatewayType	LoRaWAN	El tipo de gateway inalámbrico, cuando el recurso es <code>WirelessGateway</code> , que siempre es LoRaWAN.
wirelessDeviceType	LoRaWAN o Sidewalk	El tipo de dispositivo inalámbrico, cuando el recurso es <code>WirelessDevice</code> , que puede ser LoRaWAN o Sidewalk.
wirelessGatewayId	-	El identificador de la puerta de enlace inalámbrica, cuando el recurso es <code>WirelessGateway</code> .
wirelessDeviceId	-	El identificador del dispositivo inalámbrico, cuando el recurso es <code>WirelessDevice</code> .
event	Join, Rejoin, Registration y Certificate	<p>El tipo de evento que se registra, que depende de si el recurso que está registrando es un dispositivo inalámbrico o una puerta de enlace inalámbrica. Para obtener más información, consulte Vista CloudWatch AWS IoT Wireless Entradas de registro de en (p. 1232).</p>

AWS IoT Wireless API de registro

Puede utilizar las siguientes acciones de la API de para configurar el registro de recursos. La tabla también muestra una política de IAM de ejemplo que debe crear para utilizar las acciones de la API. En la siguiente sección se describe cómo utilizar las API de para configurar los niveles de registro de sus recursos.

Registro de las acciones de la API

Nombre de API	Descripción	Ejemplo de política de IAM
Obtener niveles de registro por tipos de recursos	Devuelve los niveles de registro predeterminados actuales o niveles de registro por tipos de recursos, que pueden incluir opciones de registro para dispositivos inalámbricos o puertas de enlace inalámbricas.	<pre>{ "Version": "2012-10-17", "Statement": [{ "Effect": "Allow", "Action": ["iotwireless:GetLogLevelByResourceTypes"] }] }</pre>

Nombre de API	Descripción	Ejemplo de política de IAM
		<pre>], "Resource": ["*"] } }</pre>
Obtener nivel de registro de recursos	Devuelve la modificación de nivel de registro de un identificador de recurso y un tipo de recurso determinados. El recurso puede ser un dispositivo inalámbrico o una puerta de enlace inalámbrica.	<pre>{ "Version": "2012-10-17", "Statement": [{ "Effect": "Allow", "Action": ["iotwireless:GetResourceLogLevel"], "Resource": ["arn:aws:iotwireless:us-east-1:123456789012:WirelessDevice/012bc5ab12-cd3a-d00e-1f0e20c1204a", ...] }] }</pre>

Nombre de API	Descripción	Ejemplo de política de IAM
Nivel de registro de recursos Put	<p>Establece la modificación a nivel de registro de un identificador de recurso y tipo de recurso determinados. El recurso puede ser una puerta de enlace inalámbrica o un dispositivo inalámbrico.</p> <p>Note</p> <p>Esta API tiene un límite de 200 sobrescrituras de nivel de registro por cuenta.</p>	<pre>{ "Version": "2012-10-17", "Statement": [{ "Effect": "Allow", "Action": ["iotwireless:PutResourceLogLevel"], "Resource": ["arn:aws:iotwireless:us-east-1:123456789012:WirelessDevice/012bc5ab12-cd3a-d00e-1f0e20c1204a", ...] }] }</pre>
Restablecer todos los niveles de registro de recursos	<p>Elimina las modificaciones de nivel de registro de todos los recursos, que incluyen puertas de enlace inalámbricas y dispositivos inalámbricos.</p> <p>Note</p> <p>Esta API no afecta a los niveles de registro que se establecen mediante el <code>UpdateLogLevelsByResourceTypesAPI</code>.</p>	<pre>{ "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/*"] }] }</pre>

Nombre de API	Descripción	Ejemplo de política de IAM
Restablecer nivel de registro de recursos	Elimina la modificación de nivel de registro de un identificador de recurso y tipo de recurso determinados. El recurso puede ser una puerta de enlace inalámbrica o un dispositivo inalámbrico.	<pre>{ "Version": "2012-10-17", "Statement": [{ "Effect": "Allow", "Action": ["iotwireless:ResetResourceLogLevel"], "Resource": ["arn:aws:iotwireless:us-east-1:123456789012:WirelessDevice/012bc5ab12-cd3a-d00e-1f0e20c1204a", ...] }] }</pre>
Actualización de los niveles de registro por tipos de recursos	<p>Defina el nivel de registro predeterminado o niveles de registro por tipos de recursos. Puede utilizar esta API para opciones de registro de dispositivos inalámbricos o puertas de enlace inalámbricas y controlar los mensajes de registro que se mostrarán en CloudWatch.</p> <p>Note</p> <p>Los eventos son opcionales y el tipo de evento está vinculado al tipo de recurso. Para obtener más información, consulte Eventos y tipos de recursos (p. 1233).</p>	<pre>{ "Version": "2012-10-17", "Statement": [{ "Effect": "Allow", "Action": ["iotwireless:UpdateLogLevelsByResourceType"], "Resource": ["*"] }] }</pre>

Configurar los niveles de registro de recursos mediante la CLI

En esta sección se describe cómo configurar niveles de registro para AWS IoT Wirelessrecursos mediante la API o AWS CLI.

Antes de utilizar la CLI:

- Asegúrese de haber creado la política de IAM para la API para la que desea ejecutar el comando CLI, como se ha descrito anteriormente.

- Necesita el nombre de recurso de Amazon (ARN) del rol que desea utilizar. Si necesita crear un rol para utilizar en el registro, consulte[Creación de una política y un rol de registro paraAWS IoT Wireless \(p. 1221\)](#).

Motivos para usarAWS CLI

De forma predeterminada, si crea el rol de IAM,`IoTWirelessLogsRole`, según se describe en[Creación de una política y un rol de registro paraAWS IoT Wireless \(p. 1221\)](#), verás CloudWatch inicios de sesión en elAWS Management Console que tienen un nivel de registro predeterminado de`ERROR`. Para cambiar el nivel de registro predeterminado de todos los recursos o de recursos específicos, utilice laAWS IoT Wirelessregistro de API o CLI.

Describe cómo utilizar elAWS CLI

Las acciones de la API se pueden clasificar en los siguientes tipos en función de si desea configurar niveles de registro para todos los recursos o para recursos específicos:

- Acciones de `APIGetLogLevelByResourceTypes`y`UpdateLogLevelByResourceTypes`puede recuperar y actualizar los niveles de registro de todos los recursos de su cuenta que sean de un tipo específico, como una puerta de enlace inalámbrica o un dispositivo LoRaWan o Sidewalk.
- Acciones de `APIGetResourceLogLevel`,`PutResourceLogLevel`,
`yResetResourceLogLevel`puede recuperar, actualizar y restablecer los niveles de registro de los recursos individuales especificados mediante un identificador de recursos.
- Acción de la API`ResetAllResourceLogLevels`restablece la modificación de nivel de registro en`null`para todos los recursos para los que ha especificado una modificación a nivel de registro mediante la`PutResourceLogLevel`API.

Para usar la CLI para configurar el registro específico de recursos para AWS IoT

Note

También puede realizar este procedimiento con la API utilizando los métodos de la API de AWS que corresponden a los comandos CLI que se muestran aquí.

1. De forma predeterminada, todos los recursos tienen el nivel de registro establecido en`ERROR`. Para establecer los niveles de registro predeterminados o los niveles de registro por tipos de recursos para todos los recursos de su cuenta, utilice la`update-log-levels-by-resource-types`comando. En el siguiente ejemplo se muestra cómo crear un archivo JSON,`Input.json`proporcionarlo como entrada para el comando de la CLI. Puede utilizar este comando para deshabilitar selectivamente el registro o anular el nivel de registro predeterminado para tipos específicos de recursos y eventos.

```
{  
    "DefaultLogLevel": "INFO",  
    "WirelessDeviceLogOptions":  
    [  
        {  
            "Type": "Sidewalk",  
            "LogLevel": "INFO",  
            "Events":  
            [  
                {  
                    "Event": "Registration",  
                    "LogLevel": "DISABLED"  
                }  
            ]  
        },  
        {  
            "Type": "LoRaWAN",  
            "LogLevel": "INFO",  
            "Events":  
            [  
                {  
                    "Event": "Registration",  
                    "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"  
        }  
      ]  
  }  
]
```

donde:

WirelessDeviceLogOptions

Lista de opciones de registro de un dispositivo inalámbrico. Cada opción de registro incluye el tipo de dispositivo inalámbrico (Sidewalk o LoraWAN) y una lista de opciones de registro de eventos de dispositivos inalámbricos. Cada opción de registro de eventos de dispositivo inalámbrico puede incluir opcionalmente el tipo de evento y su nivel de registro.

WirelessGatewayLogOptions

Lista de opciones de registro de una puerta de enlace inalámbrica. Cada opción de registro incluye el tipo de puerta de enlace inalámbrica (LoRaWAN) y una lista de opciones de registro de eventos de puerta de enlace inalámbrica. Cada opción de registro de eventos de puerta de enlace inalámbrica puede incluir opcionalmente el tipo de evento y su nivel de registro.

DefaultLogLevel

El nivel de registro que se debe utilizar en todos sus recursos. Los valores válidos son **ERROR**, **INFO** y **DISABLED**. El valor predeterminado es **INFO**.

LogLevel

El nivel de registro que desea utilizar para tipos de recursos y eventos individuales. Estos niveles de registro invalidan el nivel de registro predeterminado, como el nivel de registro **INFO** para la puerta de enlace LoraWAN y niveles de registro **DISABLED** y **ERROR** para los dos tipos de eventos.

Ejecute el siguiente comando para proporcionar el `input.json` archivo como entrada al comando. Este comando no produce ningún resultado.

```
aws iotwireless update-log-levels-by-resource-types \
--cli-input-json Input.json
```

Si desea quitar las opciones de registro de dispositivos inalámbricos y puertas de enlace inalámbricas, ejecute el siguiente comando.

```
{
    "DefaultLogLevel": "DISABLED",
    "WirelessDeviceLogOptions": [],
    "WirelessGatewayLogOptions": []
}
```

2. El comando `update-log-levels-by-resource-types` no devuelve ningún resultado. Use [get-log-levels-by-resource-types](#) para recuperar información de registro específica de recursos. El comando devuelve el nivel de registro predeterminado y las opciones de registro del dispositivo inalámbrico y la puerta de enlace inalámbrica.

Note

La `get-log-levels-by-resource-types` no puede recuperar directamente los niveles de registro en el CloudWatch consola de . Puede utilizar el `get-log-levels-by-resource-types` para obtener la información de nivel de registro más reciente que haya especificado para sus recursos mediante el `update-log-levels-by-resource-types` comando.

```
aws iotwireless get-log-levels-by-resource-types
```

Cuando ejecuta el siguiente comando, devuelve la información de registro más reciente que especificó con `update-log-levels-by-resource-types`. Por ejemplo, si quita las opciones de registro de dispositivos inalámbricos, ejecuta `get-log-levels-by-resource-types` devolverá este 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 los niveles de registro de puertas de enlace inalámbricas individuales o recursos de dispositivos inalámbricos, utilice los siguientes comandos de CLI:

- [put-resource-log-level](#)
- [get-resource-log-level](#)

- [reset-resource-log-level](#)

Para ver un ejemplo de cuándo utilizar estas CLI, supongamos que tiene un gran número de dispositivos inalámbricos o puertas de enlace en su cuenta que se están registrando. Si desea solucionar errores solo en algunos de sus dispositivos inalámbricos, puede deshabilitar el registro de todos los dispositivos inalámbricos configurando la `DefaultLogLevel` a `DISABLED`, y utilice el `put-resource-log-level` para configurar el `LogLevel` a `ERROR` solo para los dispositivos de su cuenta.

```
aws iotwireless put-resource-log-level \
    --resource-identifier
    --resource-type WirelessDevice
    --log-level ERROR
```

En este ejemplo, el comando establece el nivel de registro en `ERROR` solo para el recurso de dispositivo inalámbrico especificado y los registros de todos los demás recursos están deshabilitados. Este comando no produce ningún resultado. Para recuperar esta información y verificar que se han establecido los niveles de registro, utilice el `get-resource-log-level` comando.

4. En el paso anterior, después de depurar el problema y resolver el error, puede ejecutar la `reset-resource-log-level` para restablecer el nivel de registro de ese recurso a `null`. Si utilizó el `put-resource-log-level` para configurar la anulación de nivel de registro para más de un dispositivo inalámbrico o recurso de puerta de enlace, como para errores de solución de problemas en varios dispositivos, puede restablecer las anulaciones de nivel de registro de nuevo `null` para todos esos recursos utilizando el `reset-all-resource-log-levels` comando.

```
aws iotwireless reset-all-resource-log-levels
```

Este comando no produce ningún resultado. Para recuperar la información de registro de los recursos, ejecute el `get-resource-log-level` comando.

Pasos siguientes

Ha aprendido a crear el rol de registro y utilizar el AWS IoT Wireless API para configurar el registro para su AWS IoT Wireless de AWS. A continuación, para obtener más información sobre cómo supervisar las entradas de registro, vaya a [MonitorAWS IoT Wireless con CloudWatch Registros \(p. 1231\)](#).

MonitorAWS IoT Wireless con CloudWatch Registros

AWS IoT Wireless tiene más de 50 CloudWatch Entradas de registro que están habilitadas de forma predeterminada. Cada entrada de registro describe el tipo de evento, el nivel de registro y el tipo de recurso. Para obtener más información, consulte [AWS IoT Wireless recursos y niveles de registros \(p. 1223\)](#).

Cómo monitorizar su AWS IoT Wireless recursos

Cuando el registro está habilitado para AWS IoT Wireless, AWS IoT Wireless envía eventos de progreso acerca de cada mensaje a medida que pasa de sus dispositivos a AWS IoT de vuelta. Por defecto, AWS IoT Wireless las entradas de registro tienen un nivel de error de registro predeterminado. Cuando habilita el registro tal y como se describe en [Creación de una política y un rol de registro para AWS IoT Wireless \(p. 1221\)](#), verás mensajes en el CloudWatch consola que tiene un nivel de registro predeterminado de `ERROR`. Al utilizar este nivel de registro, los mensajes solo mostrarán información de error de todos los dispositivos inalámbricos y recursos de puerta de enlace que esté utilizando.

Si desea que los registros muestren información adicional, como los que tienen un nivel de registro de INFO o deshabilitar los registros de algunos de sus dispositivos y mostrar mensajes de registro solo para algunos de sus dispositivos, puede utilizar el AWS IoT Wireless API de registro. Para obtener más información, consulte [Configurar los niveles de registro de recursos mediante la CLI \(p. 1227\)](#).

También puede crear expresiones de filtro para mostrar solo los mensajes necesarios.

Antes de poder ver AWS IoT Wireless registros de en la consola de

Para hacer aws/iotwirelessEl grupo de registros aparece en la consola de CloudWatch; debe haber hecho lo siguiente.

- Registro habilitado AWS IoT Wireless. Para obtener más información sobre cómo habilitar el inicio de sesión en AWS IoT Wireless, consulte [Configuración del registro para AWS IoT Wireless \(p. 1221\)](#).
- Se han escrito algunas entradas de registro realizando AWS IoT Wireless operaciones.

Para crear y utilizar expresiones de filtro de forma más eficaz, le recomendamos que pruebe a utilizar CloudWatch perspectivas tal y como se describe en los siguientes temas. También te recomendamos que sigas los temas en el orden en que se presentan aquí. Esto le ayudará a utilizar CloudWatch Grupos de registros primero para obtener información sobre los distintos tipos de recursos, sus tipos de eventos y niveles de registro que puede utilizar para ver las entradas de registro en la consola. A continuación, puede aprender a crear expresiones de filtro mediante CloudWatch Insights para obtener más información útil de sus recursos.

Temas

- [Vista CloudWatch AWS IoT Wireless Entradas de registro de en \(p. 1232\)](#)
- [Usar CloudWatch Perspectivas para filtrar registros para AWS IoT Wireless \(p. 1238\)](#)

Vista CloudWatch AWS IoT Wireless Entradas de registro de en

Después de configurar el registro para AWS IoT Wireless según se describe en [Creación de una política y un rol de registro para AWS IoT Wireless \(p. 1221\)](#) y escribimos algunas entradas de registro, puede ver las entradas de registro en el CloudWatch consola siguiendo estos pasos.

Visualización de AWS IoT inicios de sesión en el CloudWatch Consola de grupos de registros

En el navegador [Consola de CloudWatch](#), CloudWatch los registros de aparecen en un grupo de registros denominado aws/iotwireless. Para obtener más información acerca de CloudWatch Registros, consulte [Registros de CloudWatch](#).

Para ver los AWS IoT inicios de sesión en el CloudWatch consola

Vaya a la [Consola de CloudWatch](#) y elige Grupos de registros en el panel de navegación.

1. En el navegador Filtro cuadro de texto, introduzca /aws/iotwireless y luego seleccione la /aws/iotwireless Grupo de registros.
2. Para ver una lista completa de los AWS IoT Wireless registros generados para tu cuenta, elige Buscar en todo. Para ver un flujo de registro individual, seleccione el icono de ampliar.
3. Para filtrar las secuencias de registro, también puede escribir una consulta en el Filtrado eventos cuadro de texto. Estas son algunas consultas que puede probar:

- { \$.logLevel = "ERROR" }

Utilice este filtro para buscar todos los registros que tengan un nivel de registro de ERROR y puede ampliar los flujos de errores individuales para leer los mensajes de error, lo que le ayudará a resolverlos.

- { \$.resource = "WirelessGateway" }

Encuentre todos los registros de la wirelessGateway recurso independientemente del nivel de registro.

- { \$.event = "CUPS_Request" && \$.logLevel = "ERROR" }

Busque todos los registros que tengan un tipo de evento de CUPS_Request y un nivel de registro de ERROR.

Eventos y tipos de recursos

En la tabla siguiente se muestran los distintos tipos de eventos para los que verás entradas de registro. Los tipos de eventos también dependen de si el tipo de recurso es un dispositivo inalámbrico o una puerta de enlace inalámbrica. Puede utilizar el nivel de registro predeterminado para los recursos y tipos de eventos o anular el nivel de registro predeterminado especificando un nivel de registro para cada uno de ellos.

Tipos de eventos basados en los recursos utilizados

Recurso	Tipo de recurso	Tipo de evento	
Puerta de enlace inalámbrica	LoRaWAN	<ul style="list-style-type: none">• Tups_request• Certificate	
Dispositivo inalámbrico	LoRaWAN	<ul style="list-style-type: none">• Join• Volver a unirse• Uplink_Data• Downlink_Data	
Dispositivo inalámbrico	Acera	<ul style="list-style-type: none">• Registro• Uplink_Data• Downlink_Data	

El tema siguiente contiene más información sobre estos tipos de eventos y las entradas de registro de puertas de enlace inalámbricas y dispositivos inalámbricos.

Temas

- [Entradas de registro para puertas de enlace inalámbricas y recursos de dispositivos inalámbricos \(p. 1233\)](#)

Entradas de registro para puertas de enlace inalámbricas y recursos de dispositivos inalámbricos

Después de habilitar el registro, podrá ver las entradas de registro de sus puertas de enlace inalámbricas y dispositivos inalámbricos. En la siguiente sección se describen los distintos tipos de entradas de registro en función de los tipos de recursos y eventos.

Entradas de registro de gateway inalámbrica

En esta sección se muestran algunas de las entradas de registro de ejemplo de los recursos de la puerta de enlace inalámbrica que verás en el[Consola de CloudWatch](#). Estos mensajes de registro pueden tener un tipo de evento comoCUPS_RequestoCertificatey se puede configurar para mostrar un nivel de registro deINFO,ERROR, o bienDISABLEDen el nivel de recursos o en el nivel de evento. Si desea ver solo información de errores, establezca el nivel de registro enERROR. El mensaje de enERRORla entrada de registro contendrá información sobre por qué ha fallado.

Las entradas de registro de su recurso de puerta de enlace inalámbrica se pueden clasificar en función de los siguientes tipos de eventos:

- Tups_request

La LoRa Basics Station que se ejecuta en la puerta de enlace envía periódicamente una solicitud al Servidor de configuración y actualización (CUPS) para obtener actualizaciones. Para este tipo de evento, si establece el nivel de registro enINFOal configurar la CLI para el recurso de puerta de enlace inalámbrica y, a continuación, en los registros:

- Si el evento se realiza correctamente, verá los mensajes de registro que tienenlogLeveldeINFO. Los mensajes incluirán detalles sobre la respuesta de CUPS enviada a la puerta de enlace y los detalles de la puerta de enlace. A continuación se muestra un ejemplo de esta entrada de registro. Para obtener más información sobre ellogLevely otros campos de la entrada de log, consulte[AWS IoT Wirelessrecursos y niveles de registros \(p. 1223\)](#).

```
{  
    "timestamp": "2021-05-13T16:56:08.853Z",  
    "resource": "WirelessGateway",  
    "wirelessGatewayId": "5da85cc8-3361-4c79-8be3-3360fb87abda",  
    "wirelessGatewayType": "LoRaWAN",  
    "gatewayEui": "feffff0000000e2",  
    "event": "CUPS_Request",  
    "logLevel": "INFO",  
    "message": "Sending CUPS response of total length 3213 to GatewayEui:  
feffff0000000e2 with TC Credentials,"  
}
```

- Si se produce un error, verás entradas de registro que tienenunlogLeveldeERRORy los mensajes incluirán detalles acerca del error. Ejemplos de cuándo puede producirse un error en elCUPS_Requestevento incluye: falta CUPS CRC, discordancia en el Uri TC de la puerta de enlace conAWS IoT Wireless, faltaIoTWirelessGatewayCertManagerRole, o no se puede obtener un registro de puerta de enlace inalámbrica. A continuación se muestra una entrada de registro CRC que falta. Para resolver el error, compruebe la configuración de la puerta de enlace para comprobar que ha introducido el CRC de CUPS correcto.

```
{  
    "timestamp": "2021-05-13T16:56:08.853Z",  
    "resource": "WirelessGateway",  
    "wirelessGatewayId": "5da85cc8-3361-4c79-8be3-3360fb87abda",  
    "wirelessGatewayType": "LoRaWAN",  
    "gatewayEui": "feffff0000000e2",  
    "event": "CUPS_Request",  
    "logLevel": "ERROR",  
    "message": "The CUPS CRC is missing from the request. Check your gateway setup and  
enter the CUPS CRC,"  
}
```

- Certificate

Estas entradas de registro le ayudarán a comprobar si la puerta de enlace inalámbrica presentó el certificado correcto para autenticar la conexión aAWS IoT. Para este tipo de evento, si establece el nivel

de registro enINFOal configurar la CLI para el recurso de puerta de enlace inalámbrica y, a continuación, en los registros:

- Si el evento se realiza correctamente, verá los mensajes de registro que tienenlogLeveldeINFO. Los mensajes incluirán detalles sobre el ID de certificado y el identificador de puerta de enlace inalámbrica. A continuación se muestra un ejemplo de esta entrada de registro. Para obtener más información sobre ellogLevely otros campos de la entrada de log, consulte[AWS IoT Wirelessrecursos y niveles de registros \(p. 1223\)](#).

```
{  
    "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)"  
}
```

- Si se produce un error, verás entradas de registro que tienenunlogLeveldeERRORy los mensajes incluirán detalles acerca del error. Ejemplos de cuándo puede producirse un error en elCertificateincluye un identificador de certificado no válido, un identificador de puerta de enlace inalámbrica o una discrepancia entre el identificador de puerta de enlace inalámbrica y el identificador de certificado. El siguiente ejemplo muestra unERRORdebido a un identificador de puerta de enlace inalámbrico no válido. Para resolver el error, compruebe los identificadores de puerta de enlace.

```
{  
    "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 registro de dispositivos inalámbricos

En esta sección se muestran algunas de las entradas de registro de ejemplo de los recursos de su dispositivo inalámbrico que verá en el[Consola de CloudWatch](#). El tipo de evento de estos mensajes de registro depende de si está utilizando un dispositivo LoRaWan o Sidewalk. Cada recurso o tipo de evento de dispositivo inalámbrico se puede configurar para mostrar un nivel de registro deINFO,ERROR, o bienDISABLED.

Note

Su solicitud no debe contener metadatos inalámbricos LoRaWan y Sidewalk al mismo tiempo. Para evitar unERRORentrada de registro para este escenario, especifique datos inalámbricos LoRaWan o Sidewalk.

Entradas de registro de dispositivos LoRaWAN

Las entradas de registro de su dispositivo inalámbrico LoraWAN se pueden clasificar en función de los siguientes tipos de eventos:

- **Join y Rejoin**

Cuando añades un dispositivo LoRaWAN y lo conectas aAWS IoT Wireless, antes de que el dispositivo pueda enviar datos de enlace ascendente, debe completar un proceso denominado [activationojoin procedure](#). Para obtener más información, consulte [Añade tu dispositivo inalámbrico aAWS IoT Core for LoRaWAN \(p. 1130\)](#).

Para este tipo de evento, si establece el nivel de registro en `INFO` al configurar la CLI para el recurso de puerta de enlace inalámbrica y, a continuación, en los registros:

- Si el evento se realiza correctamente, verá los mensajes de registro que tienen `logLevel` de `INFO`. Los mensajes incluirán detalles sobre el estado de su solicitud de afiliación o de reincorporación. A continuación se muestra un ejemplo de esta entrada de registro. Para obtener más información sobre el `logLevel` y otros campos de la entrada de log, consulte [AWS IoT Wireless recursos y niveles de registros \(p. 1223\)](#).

```
{  
    "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"  
}
```

- Si se produce un error, verás entradas de registro que tienen `logLevel` de `ERROR` y los mensajes incluirán detalles acerca del error. Ejemplos de cuándo puede producirse un error en el `Join` y `Rejoin` los eventos incluyen una configuración de región LoRaWAN no válida o la comprobación del código de integridad de mensajes (MIC) no válida. El siguiente ejemplo muestra un error de unión debido a la comprobación MIC. Para resolver el error, comprueba si has introducido las claves raíz correctas.

```
{  
    "timestamp": "2020-11-24T01:46:50.883481989Z",  
    "resource": "WirelessDevice",  
    "wirelessDeviceType": "LoRaWAN",  
    "WirelessDeviceId": "cb4c087c-1be5-4990-8654-cacf543ee9fff",  
    "devEui": "58a0cb000020255c",  
    "event": "Join",  
    "logLevel": "ERROR",  
    "message": "invalid MIC. It's most likely caused by wrong root keys."  
}
```

- Uplink_Data y Downlink_Data

El tipo de evento `Uplink_Data` se utiliza para mensajes generados por AWS IoT Wireless cuando la carga útil se envía desde el dispositivo inalámbrico a AWS IoT. El tipo de evento `Downlink_Data` se utiliza para mensajes relacionados con los mensajes de enlace descendente que se envían desde AWS IoT al dispositivo inalámbrico.

Note

Eventos `Uplink_Data` y `Downlink_Data` aplican tanto a dispositivos LoRaWan como a Sidewalk.

Para este tipo de evento, si establece el nivel de registro en `INFO` al configurar la CLI para sus dispositivos inalámbricos, en los registros, verá:

- Si el evento se realiza correctamente, verá los mensajes de registro que tienen `logLevel` de `INFO`. Los mensajes incluirán detalles sobre el estado del mensaje de enlace ascendente o descendente que se envió y el identificador del dispositivo inalámbrico. A continuación se muestra un ejemplo

de esta entrada de registro para un dispositivo Sidewalk. Para obtener más información sobre el logLevel y otros campos de la entrada de log, consulte [AWS IoT Wireless recursos y niveles de registros \(p. 1223\)](#).

```
{  
    "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\\"}"  
}
```

- Si se produce un error, verás entradas de registro que tienen un logLevel de ERROR, y los mensajes incluirán detalles sobre el error, lo que le ayudará a resolverlo. Ejemplos de cuándo puede producirse un error en el Registration evento incluye: problemas de autenticación, solicitudes no válidas o demasiadas, no se pueden cifrar ni descifrar la carga útil o no se encuentra el dispositivo inalámbrico utilizando el ID especificado. El siguiente ejemplo muestra un error de permiso detectado durante el procesamiento de un mensaje.

```
{  
    "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 registro de dispositivos de acera

Las entradas de registro de su dispositivo Sidewalk se pueden clasificar en función de los siguientes tipos de eventos:

- Registration**

Estas entradas de registro le ayudarán a supervisar el estado de cualquier dispositivo Sidewalk en el que se esté registrando. Para este tipo de evento, si establece el nivel de registro en INFO al configurar la CLI para el recurso de su dispositivo inalámbrico, en los registros, verá mensajes de registro que tienen un logLevel de INFO y ERROR. Los mensajes incluirán detalles sobre el progreso del registro desde el principio hasta la finalización. Los mensajes de registro contendrán información sobre cómo solucionar problemas relacionados con el registro del dispositivo.

A continuación se muestra un ejemplo de un mensaje de registro con un nivel de registro de INFO. Para obtener más información sobre el logLevel y otros campos de la entrada de log, consulte [AWS IoT Wireless recursos y niveles de registros \(p. 1223\)](#).

```
{  
    "resource": "WirelessDevice",  
    "wirelessDeviceId": "8d0b2775-e19b-4b2a-a351-cb8a2734a504",  
    "wirelessDeviceType": "Sidewalk",  
    "event": "Registration",  
}
```

```
"logLevel": "INFO",
"message": "Successfully completed device registration. Amazon SidewalkId =
2000000002"
}
```

- Uplink_Data y Downlink_Data

Los tipos de eventoUplink_DatayDownlink_Datapara dispositivos Sidewalk son similares a los tipos de eventos correspondientes para dispositivos LoRaWAN. Para obtener más información, consulte laUplink_Data y Downlink_Datasección descrita anteriormente para las entradas de registro de dispositivos LoRaWAN.

Pasos siguientes

Ha aprendido a ver las entradas de registro de sus recursos y las diferentes entradas de registro que puede ver en el CloudWatch consola después de habilitar el registro paraAWS IoT Wireless. Si bien puedes crear secuencias de filtros usandoGrupos de registros, le recomendamos que utilice CloudWatch Perspectivas para crear y utilizar flujos de filtros. Para obtener más información, consulte [Usar CloudWatch Perspectivas para filtrar registros paraAWS IoT Wireless \(p. 1238\)](#).

Usar CloudWatch Perspectivas para filtrar registros paraAWS IoT Wireless

Mientras puede usar CloudWatch Registros para crear expresiones de filtro; recomendamos que utilice CloudWatch información para crear y utilizar expresiones de filtro de forma más eficaz en función de la aplicación.

Le recomendamos que utilice por primera vez CloudWatch Grupos de registrospara obtener información sobre los distintos tipos de recursos, sus tipos de eventos y niveles de registro que puede utilizar para ver las entradas de registro en la consola. Puede utilizar los ejemplos de algunas expresiones de filtro de esta página como referencia para crear sus propios filtros para suAWS IoT Wirelessde AWS.

Visualización deAWS IoTinicios de sesión en el CloudWatch Consola de Logs insights

En el navegador[Consola de CloudWatch](#), CloudWatch los registros de aparecen en un grupo de registros denominado/aws/iotwireless. Para obtener más información acerca de CloudWatch Registros, consulte[Registros de CloudWatch](#).

Para ver losAWS IoTinicios de sesión en el CloudWatch consola

Vaya a la [.Consola de CloudWatch](#)y eligeLogs Insightsen el panel de navegación.

1. En el navegadorFiltrocuadro de texto, introduzca/**aws/iotwireless**y luego seleccione la/aws/iotwirelessLogs Insights.
2. Para ver una lista completa de grupos de registros, elijaSeleccione grupo (s) de registro (s). Para ver los grupos de registros deAWS IoT Wireless, elige/aws/iotwireless.

Ahora puede empezar a introducir consultas para filtrar los grupos de registros. Las siguientes secciones contienen algunas consultas útiles que le ayudarán a obtener información sobre las métricas de recursos.

Cree consultas útiles para filtrar y obtener información paraAWS IoT Wireless

Puede utilizar expresiones de filtro para mostrar información de registro útil adicional con CloudWatch Conocimientos. A continuación se muestran algunas consultas de ejemplo:

Mostrar solo registros de tipos de recursos específicos

Puede crear una consulta que le ayudará a mostrar registros solo de tipos de recursos específicos, como una puerta de enlace LoRaWan o un dispositivo Sidewalk. Por ejemplo, para filtrar los registros para mostrar solo los mensajes de los dispositivos Sidewalk, puede introducir la siguiente consulta y elegirEjecutar consulta. Para guardar esta consulta, elija Save (Guardar).

```
fields @message
| filter @message like /Sidewalk/
```

Una vez ejecutada la consulta, verás los resultados en elRegistros, que muestra las marcas de hora de los registros relacionados con los dispositivos Sidewalk de tu cuenta. También verás un gráfico de barras, que mostrará la hora en que se produjeron los eventos, si se produjeron tales eventos anteriormente relacionados con tu dispositivo Sidewalk. A continuación se muestra un ejemplo si expande uno de los resultados en elRegistrospestaña. Como alternativa, si desea solucionar los errores relacionados con los dispositivos Sidewalk, puede agregar otro filtro que establezca el nivel de registro enERRORy muestra solo información de error.

Field	Value
@ingestionTime	1623894967640
@log	954314929104:/aws/iotwireless
@logStream	WirelessDevice-
Downlink_Data-715adccfb34170214ec2f6667ddfa13cb5af2c3ddfc52fb000e554a2e780bed	
@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 mensajes o eventos específicos

Puede crear una consulta que le ayude a mostrar mensajes específicos y observar cuándo se produjeron los eventos. Por ejemplo, si desea ver cuándo se envió el mensaje de enlace descendente desde el dispositivo inalámbrico LoraWAN, puede introducir la siguiente consulta y elegirEjecutar consulta. Para guardar esta consulta, elija Save (Guardar).

```
filter @message like /Downlink message sent/
```

Una vez ejecutada la consulta, verás los resultados en el Registro, que muestra las marcas de hora cuando el mensaje de enlace descendente se envió correctamente al dispositivo inalámbrico. También verás un gráfico de barras, que mostrará la hora en que se envió un mensaje de enlace descendente, si había mensajes de enlace descendente que se enviaron previamente a tu dispositivo inalámbrico. A continuación se muestra un ejemplo si expande uno de los resultados en el Registro pestaña. Alternativamente, si no se ha enviado un mensaje de enlace descendente, puede modificar la consulta para mostrar solo los resultados de cuando el mensaje no se haya enviado, de modo que pueda depurar el problema.

Field	Value
@ingestionTime	1623884043676
@log	954314929104:/aws/iotwireless
@logStream	WirelessDevice-
Downlink_Data-42d0e6d09ba4d7015f4e9756fc6c616d401cd85fe3ac19854d9fb866153c872	
@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

Pasos siguientes

Ha aprendido a usar el en CloudWatch Perspectivas para obtener más información útil mediante la creación de consultas para filtrar los mensajes de registro. Puede combinar algunos de los filtros descritos anteriormente y diseñar sus propios filtros en función del recurso que esté monitoreando. Para obtener más información sobre el uso de CloudWatch Insights, consulte [Analizar datos de registros con CloudWatch Información detallada](#).

Después de crear consultas con CloudWatch Insights, si los ha guardado, puede cargar y ejecutar las consultas guardadas según sea necesario. Alternativamente, si hace clic en el Botón de Historia en CloudWatch Logs Insights, puede ver las consultas ejecutadas anteriormente y volver a ejecutarlas según sea necesario, o modificarlas aún más creando consultas adicionales.

Notificaciones de eventos de AWS IoT Wireless

AWS IoT Wireless puede publicar mensajes para notificarle los eventos de los dispositivos LoRaWan y Sidewalk en los que se incorpora AWS IoT Core. Por ejemplo, se le puede notificar de eventos, como cuando los dispositivos Sidewalk de su cuenta se han aprovisionado o registrado.

Cómo se puede notificar a los recursos de los eventos

Las notificaciones de eventos se publican cuando se producen determinados eventos. Por ejemplo, los eventos se generan cuando se aprovisiona el dispositivo Sidewalk. Cada evento provoca que se envíe una única notificación de evento. Las notificaciones de eventos se publican a través de MQTT con una carga JSON. El contenido de la carga depende del tipo de evento.

Note

Las notificaciones de eventos se publican al menos una vez. Es posible que se publiquen más de una. No se garantiza el orden de las notificaciones de eventos.

Eventos y tipos de recursos

En la tabla siguiente se muestran los distintos tipos de eventos para los que recibirás notificaciones. Los tipos de eventos dependen de si el tipo de recurso es un dispositivo inalámbrico, una puerta de enlace inalámbrica o una cuenta de Sidewalk. También puede habilitar eventos para sus recursos a nivel de recursos, que se aplica a todos los recursos de un tipo concreto o para recursos seleccionados, tal como se describe en la siguiente sección. Para obtener más información acerca de los distintos tipos de evento, consulte [Notificaciones de eventos para recursos de LoRaWAN \(p. 1247\)](#) y [Notificaciones de eventos para recursos Sidewalk \(p. 1252\)](#).

Tipos de eventos basados en recursos

Recurso	Tipo de recurso	Tipo de evento	
Dispositivo inalámbrico	LoRaWAN	Join	
	Acera	<ul style="list-style-type: none">Estado del registro de dispositivoProximidad	
Puerta de enlace inalámbrica	LoRaWAN	Estado de conexión	
Cuenta de acera	Acera	<ul style="list-style-type: none">Estado del registro de dispositivoProximidad	

Política para recibir notificaciones de eventos inalámbricos

Para recibir notificaciones de eventos, el dispositivo debe usar una política adecuada que le permita conectarse a la AWS IoT gateway del dispositivo y suscribirse a temas de eventos de MQTT. También debe suscribirse a los filtros de temas adecuados.

A continuación, mostramos un ejemplo de política necesaria para recibir notificaciones de los diversos eventos inalámbricos.

```
{  
    "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 los temas MQTT para eventos inalámbricos

Para enviarle notificaciones de eventos de sus recursos inalámbricos, AWS IoT utiliza los temas reservados MQTT que comienzan por un signo de dólar (\$). Puede publicar estos temas reservados y suscribirse a ellos. Sin embargo, no puede crear nuevos temas que comiencen por un signo de dólar.

Note

Los temas de MQTT son específicos de su cuenta de AWS. Utilice el formato `arn:aws:iotwireless:$aws-region:$AWS-account-ID:topic/Topic`. Para obtener más información, consulte [Temas MQTT \(p. 98\)](#).

Los temas MQTT reservados para dispositivos inalámbricos utilizan el siguiente formato:

- Temas de nivel de recursos

Estos temas se aplican a todos los recursos de un tipo concreto de su cuenta de AWS que has incorporado a AWS IoT Wireless.

`$aws/iotwireless/events/{eventName}/{eventType}/{resourceType}/resources`

- Temas a nivel de identificador

Estos temas se aplican a la selección de recursos de un tipo concreto en su cuenta de AWS que has incorporado a AWS IoT Wireless, especificado por el identificador de recurso.

`$aws/iotwireless/events/{eventName}/{eventType}/{resourceType}/
{resourceIdentifierType}/{resourceID}/{id}`

Para obtener más información acerca de los temas de los niveles de recursos e identificadores, consulte[Configuraciones de eventos \(p. 1244\)](#).

En la tabla siguiente se muestran ejemplos de temas de MQTT para los distintos eventos:

Temas MQTT y eventos

Evento	Tema MQTT	Notas
Estado de registro de dispositivos de acera	<ul style="list-style-type: none"> Tema de nivel de recursos <code>\$aws/iotwireless/events/device_registration_state/{eventType}/sidewalk/wireless_devices</code> Tema a nivel de identificador <code>\$aws/iotwireless/events/device_registration_state/{eventType}/sidewalk/{resourceType}/{resourceID}/{id}</code> 	<ul style="list-style-type: none"> {eventType} puede ser <code>registered</code> o <code>provisioned</code> {resourceType} puede ser <code>sidewalk_accounts</code> o <code>wireless_devices</code> {resourceID} es el <code>amazon_id</code> para <code>sidewalk_accounts</code> o <code>wireless_device_id</code>
Proximidad de acera	<ul style="list-style-type: none"> Tema de nivel de recursos <code>\$aws/iotwireless/events/proximity/{eventType}/sidewalk/wireless_devices</code> Tema a nivel de identificador <code>\$aws/iotwireless/events/proximity/{eventType}/sidewalk/{resourceType}/{resourceID}/{id}</code> 	<ul style="list-style-type: none"> {eventType} puede ser <code>beacon_discovered</code> o <code>beacon_lost</code> {resourceType} puede ser <code>sidewalk_accounts</code> o <code>wireless_devices</code> {resourceID} es el <code>amazon_id</code> para <code>sidewalk_accounts</code> o <code>wireless_device_id</code>
LoRaWAN	<ul style="list-style-type: none"> Tema de nivel de recursos <code>\$aws/iotwireless/events/join/{eventType}/lorawan/wireless_devices</code> Tema a nivel de identificador <code>\$aws/iotwireless/events/join/{eventType}/lorawan/</code> 	<ul style="list-style-type: none"> {eventType} puede ser <code>join_req_0_received</code>, <code>join_req_2_received</code> o <code>join_accepted</code> {resourceID} puede ser <code>wireless_device_id</code> o <code>dev_eui</code>

Evento	Tema MQTT	Notas
	<code>wireless_devices/{resourceID}/{id}</code>	
Estado de la conexión de gateway LoRaWAN	<ul style="list-style-type: none">Tema de nivel de recursos <code>\$aws/iotwireless/events/join/{eventType}/lorawan/wireless_gateways</code>Tema a nivel de identificador <code>\$aws/iotwireless/events/join/{eventType}/lorawan/wireless_gateways/{resourceID}/{id}</code>	<ul style="list-style-type: none">{eventType} puede ser <code>connected</code> o <code>disconnected</code>{resourceID} puede ser <code>wireless_gateway_id</code> o <code>gateway_eui</code>

Para obtener más información acerca de los diferentes eventos de, consulte [Notificaciones de eventos para recursos de LoRaWAN \(p. 1247\)](#) y [Notificaciones de eventos para recursos Sidewalk \(p. 1252\)](#).

Si te has suscrito a estos temas, recibirás una notificación cuando se publique un mensaje en uno de los temas de notificación de eventos. Para obtener más información, consulte [Temas reservados \(p. 100\)](#).

Precios de eventos inalámbricos

Para obtener más información sobre los precios de suscribirse a eventos y recibir notificaciones, consulte [AWS IoT Core de IPAM](#).

Habilitar eventos para recursos inalámbricos

Para que los suscriptores a los temas reservados puedan recibir mensajes, debe habilitar las notificaciones de eventos. Para ello, puede utilizar la AWS Management Console, o el AWS IoT Wireless API o AWS CLI.

Configuraciones de eventos

Puede configurar eventos para enviar notificaciones a todos los recursos que pertenecen a un tipo concreto o a recursos inalámbricos individuales. El tipo de recurso puede ser una puerta de enlace inalámbrica, una cuenta de socio de Sidewalk o un dispositivo inalámbrico, que puede ser un dispositivo LoRaWAN o Sidewalk. Para obtener información sobre el tipo de eventos que puede habilitar para sus dispositivos inalámbricos, consulte [Tipos de eventos para los recursos de LoRaWAN \(p. 1248\)](#) y [Tipos de eventos para recursos de Sidewalk \(p. 1252\)](#).

Todos los recursos de

Puede habilitar eventos de tal manera que todos los recursos de su cuenta de AWS que pertenecen a un tipo de recurso concreto recibe notificaciones. Por ejemplo, puede habilitar un evento que le notifique los cambios en el estado de la conexión de todas las puertas de enlace LoRaWAN a las que ha incorporado AWS IoT Core for LoRaWAN. La supervisión de estos eventos le ayudará a recibir notificaciones en casos tales como cuando ciertas puertas de enlace LoRaWAN de su flota de recursos se desconectan o si se pierde una baliza para varios dispositivos Sidewalk en su cuenta de AWS.

Recursos individuales

También puede agregar recursos individuales de LoRaWan y Sidewalk a la configuración de su evento y habilitar las notificaciones para ellos. Esto le ayudará a supervisar los recursos individuales de un tipo concreto. Por ejemplo, puede agregar dispositivos LoRaWan y Sidewalk seleccionados a su configuración y recibir notificaciones para eventos de estado de registro de dispositivos o uniones para estos recursos.

Requisitos previos

El recurso LoRaWan o Sidewalk debe tener una política adecuada que le permita recibir notificaciones de eventos. Para obtener más información, consulte [Política para recibir notificaciones de eventos inalámbricos \(p. 1242\)](#).

Habilitar notificaciones mediante elAWS Management Console

Para habilitar los mensajes de eventos de la consola de, vaya a la[Configuración](#)pestaña de laAWS IoTconsola y, a continuación, vaya a laNotificación de eventos LoRaWan y Sidewalksección.

Puede habilitar las notificaciones de todos los recursos de suCuenta de AWSque pertenecen a un tipo de recurso concreto y los supervisan.

Para habilitar las notificaciones de todos los recursos

1. En el navegadorNotificación de eventos LoRaWan y Sidewalk, vaya a la secciónTodos los recursospestaña, elijaAccióny luego elijaAdministrar eventos.
2. Activa los eventos que desea supervisar y, a continuación, elijaeventos de actualización. Si ya no quieres monitorizar ciertos eventos, eligeAccióny eligeAdministrar eventosy, a continuación, deshabilita esos eventos.

También puedes habilitar las notificaciones de recursos individuales en tuCuenta de AWSque pertenecen a un tipo de recurso concreto y los supervisan.

Para habilitar las notificaciones de recursos individuales

1. En el navegadorNotificación de eventos LoRaWan y Sidewalksección, elijaAccióny luego elijaAgregar recursos.
2. Elija los recursos y eventos para los que desea recibir notificaciones:
 - a. Elija si desea supervisar los eventos de suRecursos LoRaWANoRecursos de acera.
 - b. En función del tipo de recurso, puede elegir los eventos que desea habilitar para los recursos. A continuación, puede suscribirse a estos eventos y recibir notificaciones. Si elige:
 - Recursos LoRaWAN: Puedes habilitarunirseeventos para sus dispositivos LoraWAN oestado de conexióneventos para sus puertas de enlace LoRaWan.
 - Recursos de acera: Puedes habilitarEstado de registro de dispositivooproximidadeventos o ambos para sus cuentas de socios de Sidewalk y dispositivos Sidewalk.
3. Según el tipo de recurso y los eventos que haya elegido, seleccione los dispositivos inalámbricos o las puertas de enlace que desea supervisar. Puede seleccionar hasta 250 recursos para todos los recursos combinados.
4. ElegirEnviarpara añadir recursos.

Los recursos que agregue aparecerán con sus temas de MQTT en la pestaña del tipo de recurso en elNotificación de eventos LoRaWan y Sidewalksección de la consola de.

- LoRaWANlos eventos y eventos de los dispositivos Sidewalk aparecerán en elDispositivos inalámbricossección de la consola de.
- Estado de conexiónlos eventos de sus puertas de enlace de LoRaWan aparecerán en elPuertas de enlace inalámbricassección.
- Estado del registro de dispositivoyproximidadlos eventos de sus cuentas de Sidewalk aparecerán en elCuentas de acerapestaña.

Suscribirse a temas mediante el cliente MQTT

En función de si ha habilitado eventos para todos los recursos o para tipos de recursos individuales, los eventos que habilitó aparecerán en la consola con sus temas MQTT en elTodos los recursoso la pestaña del tipo de recurso especificado.

- Si elige uno de los temas de MQTT, puede ir al cliente MQTT para suscribirse a estos temas y recibir mensajes.
- Si has agregado varios eventos, puedes suscribirte a varios temas de eventos y recibir notificaciones para ellos. Para suscribirse a varios temas, elija sus temas y elijaAccióny luego elijaSuscribirse.

Habilitar notificaciones mediante elAWS CLI

Puede configurar eventos y agregar recursos a la configuración mediante elAWS IoT WirelessAPI o laAWS CLI.

Activar notificaciones para todos los recursos

Puede habilitar las notificaciones de todos los recursos de suCuenta de AWSque pertenecen a un tipo de recurso concreto y los supervisan mediante el[Actualización de la configuración de eventos por tipos de recursosAPI](#) o la[update-event-configuration-by-resource-typesCommand](#) de la CLI. Por ejemplo:

```
aws iotwireless update-event-configuration-by-resource-types \
--cli-input-json input.json
```

Contenido de input.json

```
{
    "DeviceRegistrationState": {
        "Sidewalk": {
            "AmazonIdEventTopic": "Enabled"
        }
    },
    "ConnectionStatus": {
        "LoRaWAN": {
            "WirelessGatewayEventTopic": "Enabled"
        }
    }
}
```

Note

Todas las comillas ("") van precedidas de barras diagonales invertidas (\).

Puede consultar la configuración del evento actual mediante una llamada a la[Obtener configuración de eventos por tipos de recursosAPI](#) o mediante la[get-event-configuration-by-resource-typesCommand](#) de la CLI. Por ejemplo:

```
aws iotwireless get-event-configuration-by-resource-types
```

Habilitar notificaciones para recursos individuales

Para agregar recursos individuales a la configuración de eventos y controlar qué eventos se publican mediante la API o la CLI, llame al[Actualizar la configuración del evento de recursosAPI](#) o utilice el[update-resource-event-configurationCommand](#) de la CLI. Por ejemplo:

```
aws iotwireless update-resource-event-configuration \
--identifier 1ffd32c8-8130-4194-96df-622f072a315f \
--identifier-type WirelessDeviceId \
--cli-input-json input.json
```

Contenido de input.json

```
{
  "Join": {
    "LoRaWAN": {
      "DevEuiEventTopic": "Disabled"
    },
    "WirelessDeviceIdEventTopic": "Enabled"
  }
}
```

Note

Todas las comillas ("") van precedidas de barras diagonales invertidas (\).

Puede consultar la configuración del evento actual mediante una llamada a la[Configuración de eventos GetResourceEventAPI](#) o mediante el[get-resource-event-configurationCommand](#) de la CLI. Por ejemplo:

```
aws iotwireless get-resource-event-configuration \
--identifier-type WirelessDeviceId \
--identifier 1ffd32c8-8130-4194-96df-622f072a315f
```

Lista de configuraciones de eventos

También puede utilizar laAWS IoT WirelessAPI o laAWS CLIp para enumerar las configuraciones de eventos en las que se ha habilitado al menos un tema de evento. Para enumerar las configuraciones, utilice el[Configuraciones de eventos de listaFuncionamiento de la API](#) o mediante el[list-event-configurationsCommand](#) de la CLI. Por ejemplo:

```
aws iotwireless list-event-configurations --resource-type WirelessDevice
```

Notificaciones de eventos para recursos de LoRaWAN

Puede utilizar elAWS Management ConsoleoAWS IoT WirelessOperaciones de API para notificarle los eventos de sus dispositivos y puertas de enlace LoraWAN. Para obtener más información acerca de las notificaciones de eventos y cómo habilitarlas, consulte[Notificaciones de eventos deAWS IoT Wireless \(p. 1241\)](#)y[Habilitar eventos para recursos inalámbricos \(p. 1244\)](#).

Tipos de eventos para los recursos de LoRaWAN

Los eventos que puede habilitar para los recursos de LoRaWAN incluyen:

- Únete a eventos que te notifican de los eventos de unión para tu dispositivo LoraWAN. Recibirás notificaciones cuando un dispositivo se une conAWS IoT for LoRaWAN, o cuando se recibe una solicitud de reincorporación de tipo 0 o tipo 2.
- Eventos de estado de conexión que le notifican cuando el estado de la conexión de la puerta de enlace LoraWAN cambia a conectada o desconectada.

Las siguientes secciones contienen más información sobre los eventos de los recursos de LoraWAN:

Temas

- [LoRaWAN \(p. 1248\)](#)
- [Eventos de estado de conexión \(p. 1250\)](#)

LoRaWAN

AWS IoT Core for LoRaWANpuede publicar mensajes para notificarle de los eventos de unión de los dispositivos LoraWAN en los que está integradoAWS IoT. Los eventos de unión le notifican cuando se recibe una solicitud de unirse o reincorporarse de tipo 0 o tipo 2 y el dispositivo se ha unido aAWS IoT Core for LoRaWAN.

Cómo funcionan los eventos de unión

Cuando incorporas tus dispositivos LoraWAN conAWS IoT Core for LoRaWAN,AWS IoT Core for LoRaWANRealiza ununirseprocedimiento para su dispositivo conAWS IoT Core for LoRaWAN. A continuación, el dispositivo se activa para su uso y puede enviar un mensaje de enlace ascendente para indicar que está disponible. Una vez que el dispositivo se haya unido, los mensajes de enlace ascendente y descendente se pueden intercambiar entre el dispositivo yAWS IoT Core for LoRaWAN. Para obtener más información acerca de la incorporación del dispositivo, consulte[Incorporación de sus dispositivos aAWS IoT Core for LoRaWAN \(p. 1129\)](#).

Puedes habilitar eventos para notificarte cuando tu dispositivo se ha unido aAWS IoT Core for LoRaWAN. También se te notificará si el evento de unión falla y cuando se recibe una solicitud de reincorporación de tipo 0 o tipo 2 y cuándo se acepta.

Activar eventos de unión de LoRaWAN

Antes de que los suscriptores de LoraWan se unan a temas reservados puedan recibir mensajes, debe habilitar las notificaciones de eventos para ellos desde elAWS Management Consoleo mediante la API o la CLI de. Puede habilitar estos eventos para todos los recursos de LoraWAN en suCuenta de AWS para recursos seleccionados. Para obtener información acerca de cómo habilitar estos eventos, consulte[Habilitar eventos para recursos inalámbricos \(p. 1244\)](#).

Formato de los temas de MQTT para eventos de LoRaWAN

Los temas MQTT reservados para dispositivos LoRaWAN utilizan el siguiente formato. Si te has suscrito a estos temas, todos los dispositivos LoraWAN registrados en tuCuenta de AWSpuede recibir la notificación:

- Temas de nivel de recursos
`$aws/iotwireless/events/{eventName}/{eventType}/lorawan/wireless_devices`
- Temas de identificadores

```
$aws/iotwireless/events/{eventName}/{eventType}/lorawan/wireless_devices/  
{resourceID}/{id}
```

Donde:

{eventName}

{eventName} debe ser join.

{eventType}

{eventType} puede ser:

- join_req_received
- rejoin_req_0_received
- rejoin_req_2_received
- join_accepted

{resourceID}

{resourceID} puede ser dev_eui wireless_device_id.

Por ejemplo, puede suscribirse a los siguientes temas para recibir una notificación de evento cuando AWS IoT Core for LoRaWAN ha aceptado una solicitud de unión de sus dispositivos.

```
$aws/iotwireless/events/join/join_accepted/lorawan/wireless_devices/  
wireless_device_id/{id}
```

También puede utilizar la+carácter comodín para suscribirse a varios temas al mismo tiempo. La+El carácter comodín coincide con cualquier cadena en el nivel que contiene el carácter, por ejemplo, en el siguiente tema:

```
$aws/iotwireless/events/join/join_req_received/lorawan/wireless_devices/  
wireless_device_id/+
```

Note

No puedes usar el carácter comodín#para suscribirse a los temas reservados. Para obtener más información acerca de los filtros de temas, consulte[Filtros de temas \(p. 99\)](#).

Para obtener más información acerca del uso del+comodín al suscribirse a temas, consulte[Filtros de temas \(p. 99\)](#).

Carga útil de mensajes para el evento de unión de LoRaWan

A continuación se muestra la carga útil del mensaje para el evento de unión de 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",
    }
}
```

La carga contiene los atributos siguientes:

eventId

Un ID de evento exclusivo generado por AWS IoT Core for LoRaWAN(cadena).

eventType

Tipo de evento que se ha producido. Puede ser uno de los siguientes valores:

- **join_req_received**: Este campo mostrará los parámetros de la EUIJoinEui o AppEui
- **rejoin_req_0_received**
- **rejoin_req_2_received**
- **join_accepted**: Este campo mostrará la NetId y DevAddr.

ID de dispositivo inalámbrico

El ID del dispositivo LoRaWAN.

timestamp

La marca de tiempo Unix de cuándo se produjo el evento.

DevEui

Identificador exclusivo del dispositivo que se encuentra en la etiqueta del dispositivo o en la documentación del dispositivo.

DevAddr y EUI (opcional)

Estos campos son la dirección del dispositivo opcional y los parámetros EUIJoinEUI o AppEUI.

Eventos de estado de conexión

AWS IoT Core for LoRaWAN puede publicar mensajes para notificarle los eventos de estado de conexión de las puertas de enlace LoraWAN en las que se incorpora AWS IoT. Los eventos de estado de conexión le notifican cuando el estado de conexión de una puerta de enlace LoraWAN cambia a conectada o desconectada.

Cómo funcionan los eventos de estado de conexión

Después de haber incorporado su puerta de entrada a AWS IoT Core for LoRaWAN, puede conectar la gateway a AWS IoT Core for LoRaWAN y verifique el estado de la conexión. Este evento le notifica cuando el estado de la conexión de la puerta de enlace cambia a conectada o desconectada. Para obtener más información sobre cómo incorporar y conectar la puerta de enlace a AWS IoT Core for LoRaWAN, consulte [Incorporación de sus gateways a AWS IoT Core for LoRaWAN \(p. 1123\)](#) y [Connect el LoRaPuerto de enlace WAN y verificar el estado de su conexión \(p. 1128\)](#).

Formato de los temas de MQTT para puertas de enlace LoRaWAN

Los temas MQTT reservados para las puertas de enlace LoRaWAN utilizan el siguiente formato. Si se ha suscrito a estos temas, entonces todas las puertas de enlace de LoraWAN registradas en su cuenta de AWS puede recibir la notificación:

- Para temas de nivel de recursos:

```
$aws/iotwireless/events/{eventName}/{eventType}/lorawan/wireless_gateways
```

- Para temas de identificadores:

```
$aws/iotwireless/events/{eventName}/{eventType}/lorawan/wireless_gateways/{resourceID}/{id}
```

Donde:

{eventName}

{eventName} debe ser connection_status.

{eventType}

{eventType} puede ser connected o disconnected.

{resourceID}

{resourceID} puede ser gateway_eui o wireless_gateway_id.

Por ejemplo, puede suscribirse a los siguientes temas para recibir una notificación de evento cuando todas las puertas de enlace se hayan conectado a AWS IoT Core for LoRaWAN:

```
$aws/iotwireless/events/connection_status/connected/lorawan/wireless_gateways/wireless_gateway_id/{id}
```

También puede utilizar la + carácter comodín para suscribirse a varios temas al mismo tiempo. El + carácter comodín coincide con cualquier cadena en el nivel que contiene el carácter, por ejemplo, en el siguiente tema:

```
$aws/iotwireless/events/connection_status/connected/lorawan/wireless_gateways/wireless_gateway_id/+
```

Note

No puedes usar el carácter comodín # para suscribirse a los temas reservados. Para obtener más información acerca de los filtros de temas, consulte [Filtros de temas \(p. 99\)](#).

Para obtener más información acerca del uso del + comodín al suscribirse a temas, consulte [Filtros de temas \(p. 99\)](#).

Carga útil de mensajes para eventos de estado de conexión

A continuación se muestra la carga útil del mensaje del evento de estado de la conexión.

```
{
    // General fields
    "eventId": "string",
    "eventType": "connected|disconnected",
    "WirelessGatewayId": "string",
    "timestamp": "timestamp",

    // Event-specific fields
    "LoRaWAN": {
        "GatewayEui": "string"
    }
}
```

La carga contiene los atributos siguientes:

eventId

Un ID de evento exclusivo generado por AWS IoT Core for LoRaWAN(cadena).

eventType

Tipo de evento que se ha producido. Puede ser `connected` o `disconnected`.

ID de puerta de enlace inalámbrica

El ID de la gateway de LoRaWan.

timestamp

La marca de tiempo Unix de cuándo se produjo el evento.

GatewayEui

Identificador exclusivo de la puerta de enlace que se encuentra en la etiqueta de puerta de enlace o la documentación de la puerta

Notificaciones de eventos para recursos Sidewalk

Puede utilizar el AWS Management Console o las operaciones de API para notificarle los eventos de sus dispositivos Sidewalk y cuentas de socios. Para obtener más información acerca de las notificaciones de eventos y cómo habilitarlas, consulte [Notificaciones de eventos de AWS IoT Wireless \(p. 1241\)](#) y [Habilitar eventos para recursos inalámbricos \(p. 1244\)](#).

Tipos de eventos para recursos de Sidewalk

Los eventos que puede habilitar para los recursos de Sidewalk incluyen:

- Eventos de dispositivo que le notifican cambios en el estado de su dispositivo Sidewalk, como cuando el dispositivo se ha registrado y está listo para usar.
- Eventos de proximidad que te avisan cuando AWS IoT Wireless recibe una notificación de Amazon Sidewalk de que se ha descubierto o perdido un faro.

Las siguientes secciones contienen más información sobre los eventos de los recursos de Sidewalk:

Temas

- [Eventos de estado de registro de dispositivos \(p. 1252\)](#)
- [Eventos de proximidad \(p. 1255\)](#)

Eventos de estado de registro de dispositivos

Los eventos de estado de registro de dispositivos publican notificaciones de eventos cuando se produce un cambio en el estado de registro del dispositivo, como cuando se ha aprovisionado o registrado un dispositivo Sidewalk. Los eventos proporcionan información sobre los diferentes estados por los que pasa el dispositivo desde el momento en que se aprovisiona hasta el momento en que se ha registrado.

Cómo funcionan los eventos de estado de registro de dispositivos

Cuando incorporas tu dispositivo Sidewalk con Amazon Sidewalk y AWS IoT Wireless, AWS IoT Wireless realiza `uncreate` y añade su dispositivo Sidewalk a su cuenta de AWS. A continuación, el

dispositivo entra en el estado aprovisionado y `eventType` se convertirá en `provisioned`. Para obtener más información acerca de la incorporación de su dispositivo, consulte [Dispositivos Sidewalk integrados con integración de Amazon Sidewalk para AWS IoT Core \(p. 1212\)](#).

Una vez que el dispositivo haya sido `provisioned`, Amazon Sidewalk realiza una `register` operación para registrar su dispositivo Sidewalk con AWS IoT Wireless. Comienza el proceso de registro, donde las claves de cifrado y sesión se configuran con AWS IoT. Cuando el dispositivo está registrado, `eventType` se convertirá en `registered`, y su dispositivo estará listo para usar.

Una vez que el dispositivo haya sido `registered`, Sidewalk puede enviar una solicitud `register` para dispositivo AWS IoT Wireless luego de satisfacer la solicitud y cambiar el estado del dispositivo de nuevo a `provisioned`. Para obtener más información acerca de los estados del dispositivo, consulte [Estado del dispositivo](#).

Activar notificaciones para eventos de estado de registro de dispositivos

Antes de que los suscriptores al estado de registro del dispositivo, los temas reservados puedan recibir mensajes, debe habilitar las notificaciones de eventos para ellos desde el AWS Management Console mediante la API o la CLI de. Puede habilitar estos eventos para todos los recursos de Sidewalk de su cuenta de AWS para recursos seleccionados. Para obtener información acerca de cómo habilitar estos eventos, consulte [Habilitar eventos para recursos inalámbricos \(p. 1244\)](#).

Formato de los temas MQTT para eventos de estado de registro de dispositivos

Para notificarle los eventos del estado de registro de dispositivos, puede suscribirse a temas reservados de MQTT que comienzan con un signo de dólar (\$). Para obtener más información, consulte [Temas MQTT \(p. 98\)](#).

Los temas MQTT reservados para eventos de estado de registro de dispositivos Sidewalk utilizan el siguiente formato:

- Para temas de nivel de recursos:

```
$aws/iotwireless/events/{eventName}/{eventType}/sidewalk/wireless_devices
```

- Para temas de identificadores:

```
$aws/iotwireless/events/{eventName}/{eventType}/sidewalk/{resourceType}/{resourceID}/{id}
```

Donde:

{eventName}

{eventName} debe ser `device_registration_state`.

{eventType}

{eventType} puede ser `provisioned` o `registered`.

{resourceType}

{resourceType} puede ser `sidewalk_accounts` o `wireless_devices`.

{resourceID}

```
{resourceID} es amazon_id para {resourceType}  
de sidewalk_accounts o wireless_device_id para {resourceType} de wireless_devices.
```

También puede utilizar la+carácter comodín para suscribirse a varios temas al mismo tiempo. La+El carácter comodín coincide con cualquier cadena en el nivel que contiene el carácter. Por ejemplo, si desea recibir una notificación de todos los tipos de evento posibles (`provisioned|registered`) y para todos los dispositivos registrados en un Amazon ID concreto, puede utilizar el siguiente filtro de temas:

```
$aws/iotwireless/events/device_registration_state/+/sidewalk/sidewalk_accounts/  
amazon_id/+
```

Note

No puedes usar el carácter comodín#para suscribirse a los temas reservados. Para obtener más información acerca de los filtros de temas, consulte[Filtros de temas \(p. 99\)](#).

Carga útil de mensajes para eventos de estado de registro de dispositivos

Una vez habilitadas las notificaciones de eventos de estado de registro de dispositivos, las notificaciones de eventos se publican a través de MQTT con una carga JSON. Estos eventos contienen la siguiente carga de ejemplo:

```
{  
    "eventId": "string",  
    "eventType": "provisioned|registered",  
    "WirelessDeviceId": "string",  
    "timestamp": "timestamp",  
  
    // Event-specific fields  
    "operation": "create|deregister|register",  
    "Sidewalk": {  
        "AmazonId": "string",  
        "SidewalkManufacturingSn": "string"  
    }  
}
```

La carga contiene los atributos siguientes:

eventId

Un ID de evento exclusivo (cadena).

eventType

Tipo de evento que se ha producido. Puede ser `provisioned` o `registered`.

ID de dispositivo inalámbrico

El identificador del dispositivo inalámbrico.

timestamp

La marca de tiempo Unix de cuándo se produjo el evento.

operación

La operación en la que se activó el evento. Los valores válidos son `create`, `register` y `deregister`.

acera

El ID de Amazon de Sidewalk o SidewalkManufacturingSn para los que desea recibir notificaciones de eventos.

Eventos de proximidad

Eventos de proximidad publican notificaciones de eventos cuando AWS IoT recibe una baliza del dispositivo Sidewalk. Cuando su dispositivo Sidewalk se acerca a Amazon Sidewalk, Amazon Sidewalk filtra las balizas que se envían desde su dispositivo a intervalos regulares y las recibe AWS IoT Wireless. AWS IoT Wireless, a continuación, le notifica de estos eventos cuando se recibe una baliza.

Cómo funcionan los eventos de proximidad

Los eventos de proximidad te avisan cuando AWS IoT recibe un faro. Sus dispositivos Sidewalk pueden emitir balizas en cualquier momento. Cuando tu dispositivo está cerca de Amazon Sidewalk, Sidewalk recibe las balizas y las reenvía a AWS IoT Wireless a intervalos de tiempo regulares. Amazon Sidewalk ha configurado este intervalo de tiempo en 10 minutos. Cuando AWS IoT Wireless recibe la baliza de Sidewalk, se te notificará del evento.

Los eventos de proximidad te avisarán cuando se descubre una baliza o cuando se pierde una baliza. Puede configurar los intervalos en los que se le notifica del evento de proximidad.

Activar notificaciones para eventos de proximidad

Antes de que los suscriptores a los temas reservados de proximidad de Sidewalk puedan recibir mensajes, debe habilitar las notificaciones de eventos para ellos desde el AWS Management Console mediante la API o la CLI de. Puede habilitar estos eventos para todos los recursos de Sidewalk de su Cuenta de AWS o para recursos seleccionados. Para obtener información acerca de cómo habilitar estos eventos, consulte [Habilitar eventos para recursos inalámbricos \(p. 1244\)](#).

Formato de los temas MQTT para eventos de proximidad

Para notificarle los eventos de proximidad, puede suscribirse a temas reservados de MQTT que comienzan con un signo de dólar (\$). Para obtener más información, consulte [Temas MQTT \(p. 98\)](#).

Los temas MQTT reservados para eventos de proximidad de Sidewalk utilizan el formato:

- Para temas de nivel de recursos:

```
$aws/iotwireless/events/{eventName}/{eventType}/sidewalk/wireless_devices
```

- Para temas de identificadores:

```
$aws/iotwireless/events/{eventName}/{eventType}/sidewalk/{resourceType}/{resourceID}/{id}
```

Donde:

{eventName}

{eventName} debe ser proximity.

{eventType}

{eventType} puede ser beacon_discovered o beacon_lost.

{resourceType}

{resourceType} puede ser sidewalk_accounts o wireless_devices.

{resourceID}

```
{resourceID} es amazon_id para {resourceType}  
de sidewalk_accounts y wireless_device_id para {resourceType} de wireless_devices.
```

También puede utilizar la+carácter comodín para suscribirse a varios temas al mismo tiempo. La+El carácter comodín coincide con cualquier cadena en el nivel que contiene el carácter. Por ejemplo, si desea recibir una notificación de todos los tipos de evento posibles (`beacon_discovered|beacon_lost`) y para todos los dispositivos registrados en un Amazon ID concreto, puede utilizar el siguiente filtro de temas:

```
$aws/iotwireless/events/proximity/+/sidewalk/sidewalk_accounts/amazon_id/+
```

Note

No puedes usar el carácter comodín#para suscribirse a los temas reservados. Para obtener más información acerca de los filtros de temas, consulte[Filtros de temas \(p. 99\)](#).

Carga útil de mensajes para eventos de proximidad

Después de habilitar las notificaciones de eventos de proximidad, los mensajes de evento se publican a través de MQTT con una carga JSON. Estos eventos contienen la siguiente carga de ejemplo:

```
{
    "eventId": "string",
    "eventType": "beacon_discovered|beacon_lost",
    "WirelessDeviceId": "string",
    "timestamp": "1234567890123",

    // Event-specific fields
    "Sidewalk": {
        "AmazonId": "string",
        "SidewalkManufacturingSn": "string"
    }
}
```

La carga contiene los atributos siguientes:

eventId

Un identificador de evento único, que es una cadena.

eventType

Tipo de evento que se ha producido. Puede ser `beacon_discovered` o `beacon_lost`.

ID de dispositivo inalámbrico

El identificador del dispositivo inalámbrico.

timestamp

La marca de tiempo Unix de cuándo se produjo el evento.

acera

El ID de Amazon de Sidewalk o SidewalkManufacturingSn para los que desea recibir notificaciones de eventos.

Integración de Alexa Voice Service (AVS) para AWS IoT

Integración de Alexa Voice Service (AVS) para AWS IoT es una nueva característica que incorpora Alexa Voice en cualquier dispositivo conectado de forma económica sin incurrir en.. AVS para AWS IoT reduce el costo y la complejidad de la integración de Alexa. Esta función utiliza AWS IoT para transferir las tareas de audio que hacen un uso intensivo de procesos informáticos y de la memoria del dispositivo a la nube. Gracias al ahorro de costos en la lista de materiales de ingeniería (eBoM), ahora los fabricantes de dispositivos pueden incorporar Alexa en dispositivos IoT con recursos limitados de manera más económica para permitir que los consumidores hablen directamente con Alexa en su hogar, oficina o habitación del hotel y disfruten así de una experiencia de sonido ambiente.

Actualmente, los dispositivos de IoT domésticos inteligentes se fabrican con microcontroladores de bajo costo (MCU) que tienen memoria limitada para ejecutar sistemas operativos en tiempo real. Anteriormente, las soluciones AVS para productos integrados de Alexa requerían costosos dispositivos basados en procesadores de aplicaciones con más de 50 MB de memoria ejecutándose en Linux o Android. Con estos costosos requisitos de hardware, resultaba prohibitivo integrar Alexa Voice en dispositivos IoT con recursos limitados. AVS para AWS IoT permite la funcionalidad integrada de Alexa en MCU, como los procesadores ARM Cortex M con menos de 1 MB de RAM integrada. Para ello, AVS transfiere las tareas de memoria y computación a un dispositivo virtual de Alexa integrado en la nube. Esto reduce el costo de eBoM hasta en un 50 %.

Para obtener más información sobre los procesadores de la serie ARM Cortex-M, consulte [ARM](#) o [Wikipedia](#). Para obtener más información acerca de los requisitos de hardware para los productos integrados de Alexa, consulte [Aumento de la CPU, la memoria y el almacenamiento del dispositivo integrado de Alexa](#) en el portal para desarrolladores de Amazon Alexa.

Note

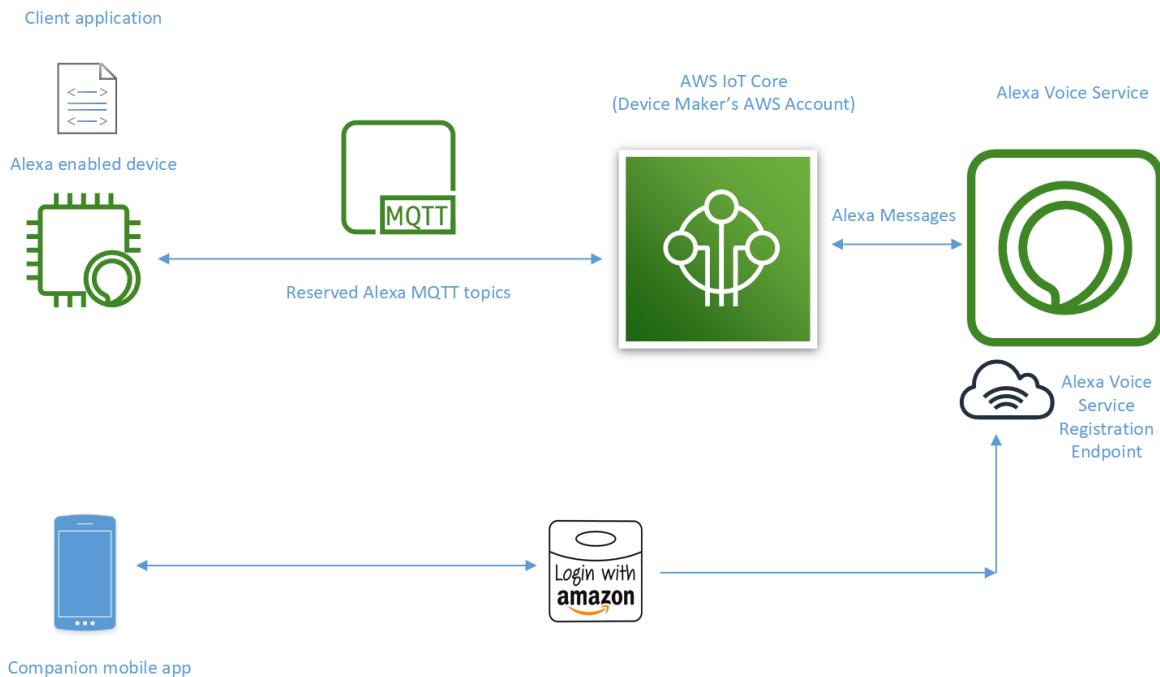
AVS para AWS IoT está disponible en todos los Regiones de AWS donde AWS IoT está disponible excepto en las regiones de China (Pekín y Ningxia). Para ver la lista actual de las Regiones de AWS, consulte la [AWSTabla de regiones](#).

AVS para AWS IoT tiene tres componentes:

- Un conjunto de temas de MQTT reservados para transferir mensajes de audio entre dispositivos compatibles con Alexa y AVS.
- Un dispositivo virtual habilitado para Alexa en la nube que transfiere las tareas relacionadas con la recuperación de medios, la descodificación de audio, la mezcla de audio y la administración del estado del dispositivo físico al dispositivo virtual.
- Un conjunto de API que permiten la recepción y el envío de mensajes a través de temas reservados, la interacción con el micrófono y el altavoz del dispositivo, y la administración del estado del dispositivo.

El siguiente diagrama ilustra cómo estos componentes funcionan juntos. También muestra cómo los fabricantes de dispositivos utilizan el servicio Login with Amazon para autenticarse en el servicio AVS.

Alexa Voice Service (AVS) Integration for AWS IoT Core



Los fabricantes de dispositivos tienen dos opciones para comenzar a utilizar la integración de AVS para AWS IoT .

- Kits de desarrollo— Los kits de desarrollo lanzados por nuestros socios facilitan el comienzo. La [NXP i.MX RT 106 A](#), [Kit de desarrollo Qualcomm Home Hub 100 para Amazon AVS](#), y [STM32 STEVAL-VOICE-U1](#) son algunos de los kits disponibles en el mercado. Puede encontrarlos en [Kits de desarrollo para AVS](#). Los kits incluyen conectividad inmediata con AWS IoT, algoritmos de audio aptos para AVS para captura de voz de largo alcance, cancelación de eco, Alexa Wake Word y AVS para código de aplicación de AWS IoT. Puede usar el código de la aplicación para crear rápidamente prototipos de un dispositivo y mover la implementación al diseño de MCU elegido para las pruebas y producción de dispositivos cuando esté listo.
- Código de aplicación personalizado del lado del dispositivo— Los desarrolladores también pueden escribir un AVS personalizado para AWS IoT mediante la API disponible públicamente. La documentación de esta API está disponible en la [página para desarrolladores de AVS](#). Puede descargar FreeRTOS y AWS IoT SDK de dispositivo desde la consola FreeRTOS (<https://console.aws.amazon.com/freertos/>) o [GitHub](#).

Para ver un ejemplo de cómo empezar a usar un kit de desarrollo, consulte [Introducción a la integración de Alexa Voice Service \(AVS\) para AWS IoT en un dispositivo NXP](#).

Introducción a la integración de Alexa Voice Service (AVS) para AWS IoT en un dispositivo NXP

Con el kit de desarrollo NXP i.MX RT106A, puede obtener una vista previa de la integración de Alexa Voice Service (AVS) para AWS IoT utilizando una cuenta de NXP preconfigurada. Después de obtener una vista

previa de la funcionalidad con la cuenta de NXP, personalice el firmware (código fuente de la aplicación) para utilizar su propia cuenta. En este tema se describen los pasos para obtener una vista previa con la cuenta preconfigurada y personalizar el dispositivo con su propia cuenta.

Temas

- [Vista previa de la integración de Alexa Voice Service \(AVS\) paraAWS IoTcon una cuenta de NXP preconfigurada \(p. 1259\)](#)
- [Interactúa con Alexa \(p. 1270\)](#)
- [Utilizar suAWSy cuentas de desarrollador de Alexa Voice Service para configurar AVS paraAWS IoT \(p. 1272\)](#)

Vista previa de la integración de Alexa Voice Service (AVS) paraAWS IoTcon una cuenta de NXP preconfigurada

Requisitos previos

Para seguir estos pasos, necesita los siguientes recursos.

- [Kit de desarrollo NXP i.MX RT106A](#)

Este kit está precargado con software que permite ambas[Configuración de cero touch \(p. 1270\)](#)y[configuración guiada por el usuario \(p. 1261\)](#).

- Un equipo Mac, Windows 7 o 10 o Linux
- Un dispositivo móvil Android o iOS
- La[Aplicación Amazon Alexa para iOS](#) o el[Aplicación Amazon Alexa para Android](#)
- Una [cuenta de Amazon Alexa](#)

Encender el kit de desarrollo

Compruebe que la caja del kit de desarrollo contenga un cable USB tipo C a doble tipo A. Conecte ambas conexiones USB-A a su equipo. Conecte el conector USB-C al kit. La configuración tendrá un aspecto similar a la siguiente imagen.



Note

Si la caja contiene una tarjeta de inicio rápido, no la tenga en cuenta y consulte estas instrucciones.

Cuando la placa tenga alimentación, el indicador LED del indicador de estado se iluminará y muestra varios colores. Se trata de indicadores de estado de las distintas fases del proceso de arranque. Los colores y la frecuencia de parpadeo indican el estado del dispositivo. El dispositivo está listo para su configuración cuando la luz indicadora de estado se vuelve azul fijo, como se muestra en la siguiente imagen.



El kit de desarrollo admite las siguientes configuraciones, según su entorno.

- [Configuración guiada por el usuario \(p. 1261\)](#): Utilice esta configuración cuando el dispositivo llegue en estado de fábrica y no cumpla las condiciones para la configuración sin contacto (ZTS).

También se utiliza la configuración guiada por el usuario cuando alguien ya ha realizado ZTS en el dispositivo. El ZTS solo puede ocurrir una vez en la vida útil de un producto.

- [Configuración táctil cero \(ZTS\) \(p. 1270\)](#): Utilice esta configuración cuando el entorno cumple las siguientes condiciones.
 - Has comprado el kit en Amazon.com.
 - No compraste el kit ni lo recibiste como regalo.
 - Ya ha instalado un dispositivo aprovisionador en la red wifi que está utilizando con el kit.

Un dispositivo aprovisionador es un dispositivo Amazon (como un Echo (tercera generación)) registrado en una cuenta de cliente de Amazon.

Para obtener una lista de los dispositivos Amazon que califican como dispositivos de aprovisionamiento, consulte [Pruebas de su dispositivo en Descripción de la configuración sin frustración](#).

- El kit se encuentra dentro del rango Bluetooth de baja energía (BLE) del dispositivo de aprovisionamiento.
- Tus credenciales de Wi-Fi están disponibles en el casillero Wi-Fi de Amazon.
- Tiene una [Habilidad de Alexa](#) vinculada a tu cuenta de Amazon.
- Ha implementado [Login with Amazon](#).

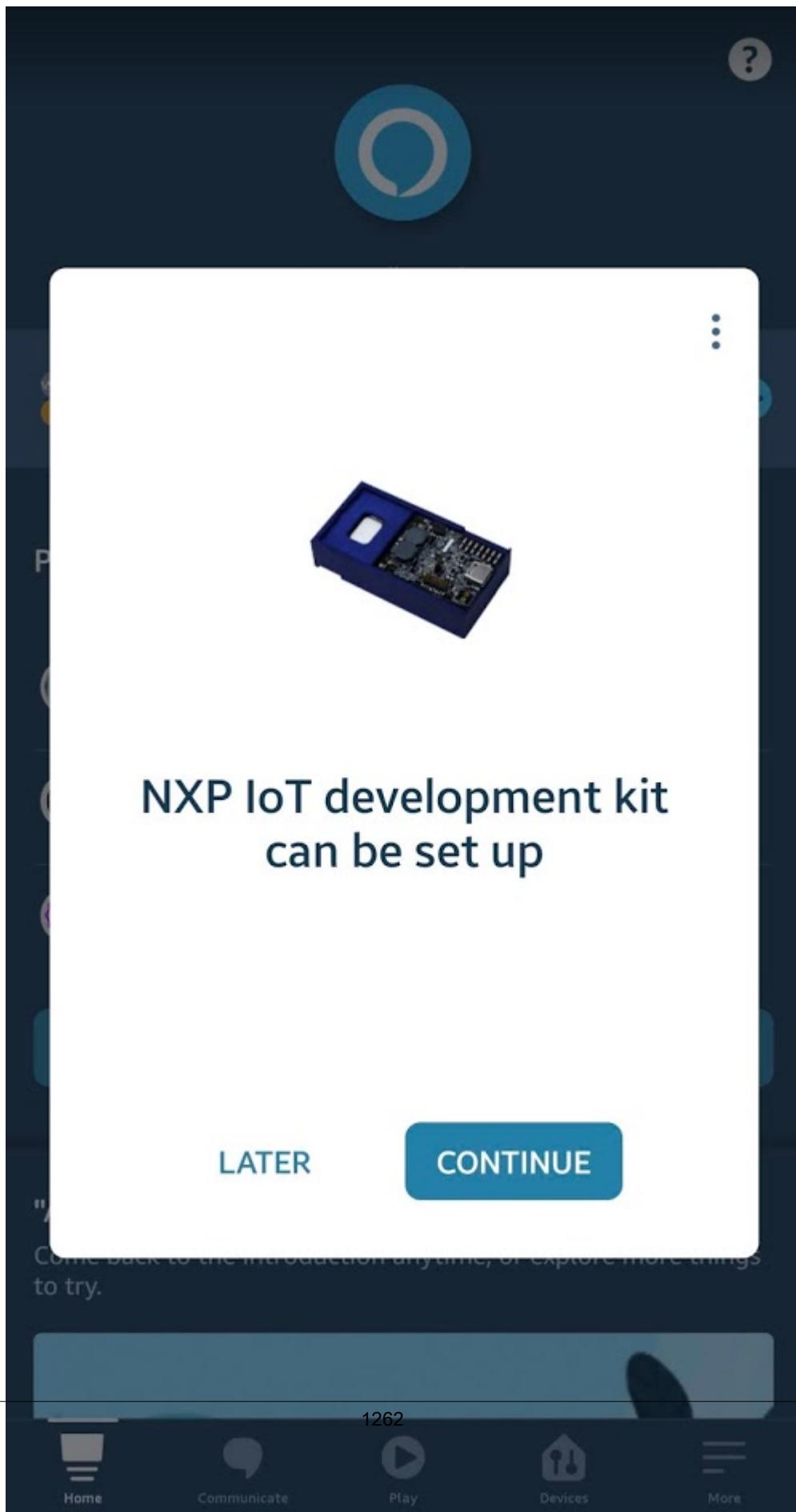
Para obtener más información acerca de este tipo de configuración, consulte [Configuración de Zero Touch](#).

Configuración guiada por el usuario

Cuando se enciende un kit que no cumple los requisitos de ZTS, espera a que se produzca una configuración guiada por el usuario a través de la aplicación Amazon Alexa de tu teléfono. Asegúrate de que la aplicación Amazon Alexa esté instalada en tu teléfono y de que los permisos de ubicación y Bluetooth estén habilitados para la aplicación.

En el siguiente procedimiento, se describe cómo realizar una configuración guiada por el usuario.

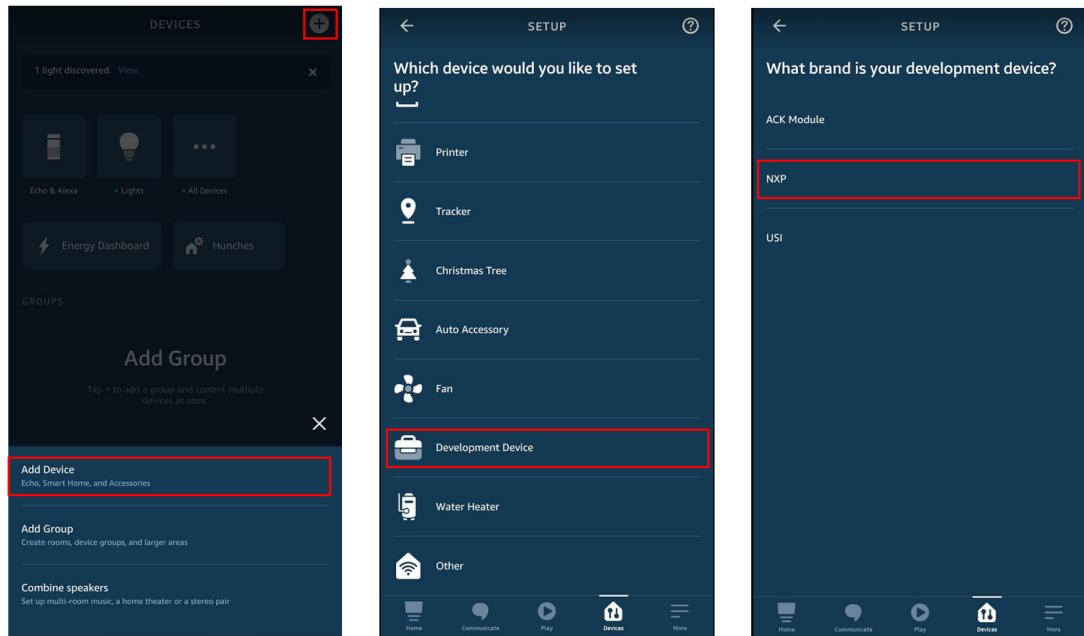
1. Abre la aplicación Alexa e inicia sesión en tu cuenta de Amazon Alexa. La aplicación detecta que un dispositivo cercano está esperando una configuración guiada por el usuario y muestra la página en la siguiente imagen. Elija Continuar.



Si elige **Posterior** si la aplicación no muestra esta página, siga los siguientes pasos para iniciar la configuración guiada por el usuario.

1. Elija el icono **Dispositivos** y luego seleccione el signo más (+) en la ventana que aparece.
2. Elegir **Agregar dispositivo**.
3. Elegir **Dispositivo de desarrollo**.
4. En la página **¿Qué marca es tu dispositivo de desarrollo?** página, elija **NXPy** luego elija **Próximo**.

Las imágenes siguientes muestran cómo aparecen en la aplicación las indicaciones descritas en estos pasos.



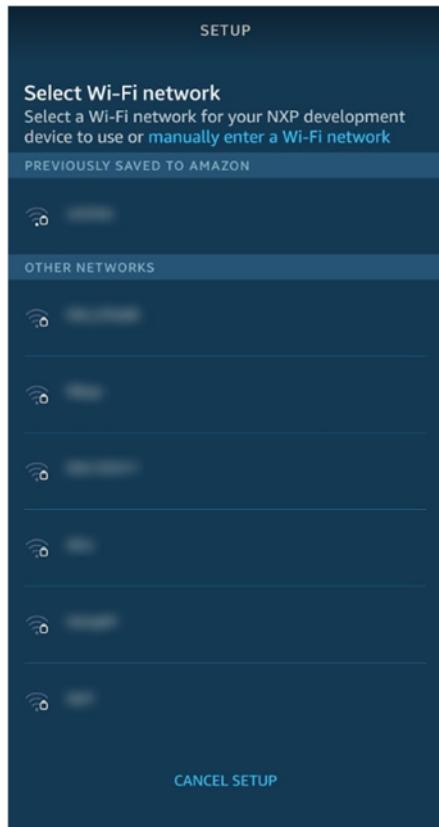
Cuando la aplicación se conecta al dispositivo, la luz indicadora de estado parpadea en naranja, como en las imágenes siguientes.



Note

Si se interrumpe la configuración guiada por el usuario (por ejemplo, si cierra la aplicación), el dispositivo vuelve al modo de detección y la luz indicadora de estado se muestra en azul fijo.

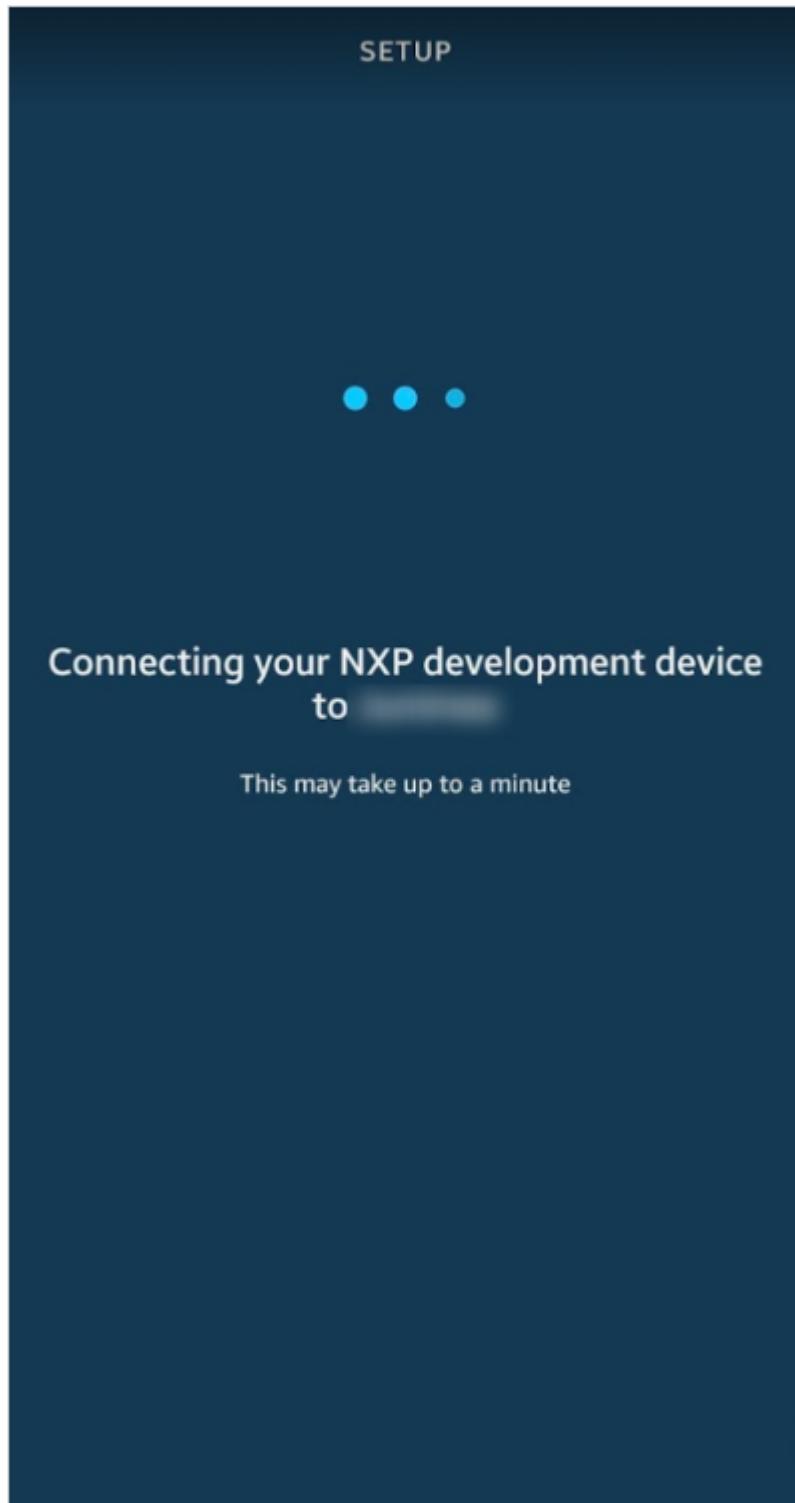
2. La aplicación pide al kit que analice el entorno en busca de redes Wi-Fi y devuelva una lista de redes que detecta. Elija la red a la que debe conectarse el dispositivo. La imagen siguiente muestra cómo aparece esta lista en la aplicación.



Note

Si ya has guardado la red seleccionada en tu cuenta de Amazon, no necesitas introducir la contraseña de Wi-Fi.

Cuando seleccionas la red Wi-Fi, la pantalla muestra el siguiente mensaje a medida que se produce el aprovisionamiento de Wi-Fi y la comunicación con los servidores de configuración: Conexión de su dispositivo de desarrollo NXP a **Nombre de red wifi**. La imagen siguiente muestra cómo se muestra esta pantalla en la aplicación.



La luz indicadora de estado sigue parpadeando en naranja hasta que se complete el registro del kit. Cuando se haya completado el registro, el dispositivo dice: «Tu dispositivo Alexa está listo». El kit se reinicia.

A continuación se describen los pasos que sigue el kit después de reiniciarse y volver a conectarse a la red Wi-Fi que ha seleccionado.

1. A medida que se reinicia, el kit vuelve a mostrar varios colores y alterna entre parpadeos y colores sólidos a medida que avanza durante el proceso de arranque.
2. A continuación, el dispositivo intenta volver a conectar a la red wifi que ha seleccionado. Al hacerlo, la luz indicadora de estado parpadea en amarillo a intervalos de 500 milisegundos (ms). Después de conectarse a la red Wi-Fi, parpadea en amarillo más rápido, a intervalos de 250 ms. Las imágenes siguientes muestran cómo aparece este parpadeo en el kit.

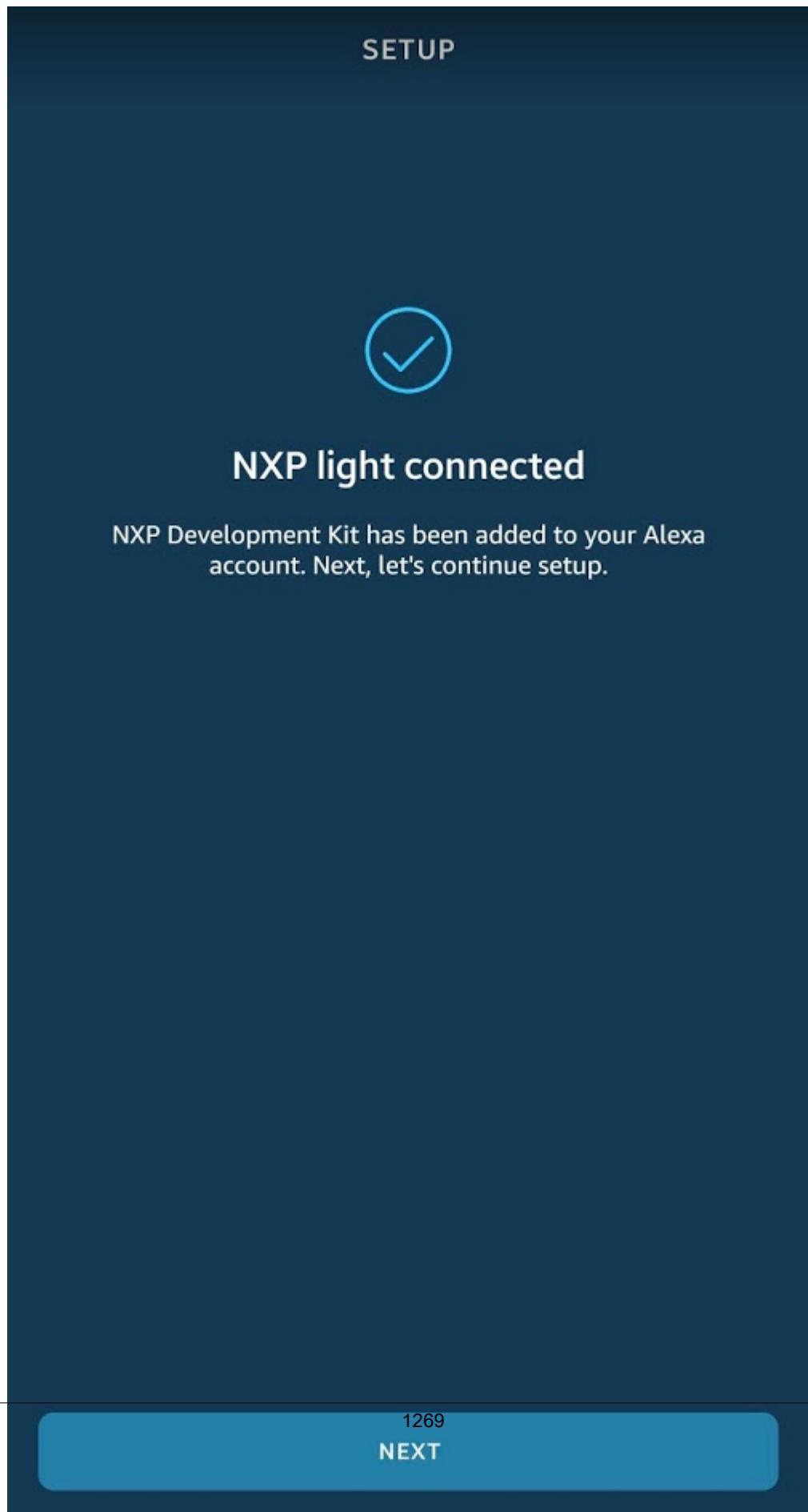


3. El kit se conecta aAWS IoT. Mientras se conecta, la luz indicadora de estado parpadea en verde a intervalos de 500 ms. Cuando el kit haya terminado de conectarse, la luz indicadora de estado parpadea en verde a intervalos de 250 ms. Las imágenes siguientes muestran cómo aparece este parpadeo en el kit.



4. El kit reproduce un sonido de timbre que indica que puedes usarlo para interactuar con Alexa.

Cuando el kit se conecta a AWS IoT, la pantalla de la siguiente imagen aparece en la aplicación.



LaLuz NXP conectada aparece en la aplicación porque el kit implementa las capacidades de hogar inteligente para un dispositivo de iluminación NXP.

Configuración táctil cero (ZTS)

Si su entorno cumple todos los requisitos previos de ZTS, el dispositivo de aprovisionamiento descubre el kit e inicia la configuración de ZTS cuando enciende el kit. La aplicación Amazon Alexa también es un aprovisionador de ZTS, por lo que abrir la aplicación Amazon Alexa también puede iniciar la configuración de ZTS.

A medida que continúa el proceso de aprovisionamiento, el estado de la luz del indicador de estado sigue los mismos patrones que los descritos en la sección de configuración guiada por el usuario. Durante el aprovisionamiento, los mensajes de registro se envían a la consola SLN-ALEXA-IOT a través de su puerto COM virtual. Una vez finalizado el aprovisionamiento, el kit reproduce el sonido del timbre que indica que puedes usarlo para interactuar con Alexa.

Note

La configuración de ZTS solo puede realizarse una vez en la vida útil de un dispositivo, incluso si lo devuelve a la configuración de fábrica.

Interactúa con Alexa

Puedes empezar a usar el kit para interactuar con Alexa haciéndole una pregunta. Incluso una pregunta simple, como «Alexa, ¿cómo está el tiempo?» pasa por varios estados a medida que Alexa procesa y responde a él.

Verás la primera indicación de que el kit está escuchando cuando dices la palabra de activación de Alexa. Cuando el kit detecta esta palabra, el kit comienza a escuchar y enviar información desde el micrófono a AVS a través de AWS IoT. La luz indicadora de estado muestra un color cian sólido, como en la imagen siguiente.



Cuando el dispositivo termine de enviar información desde el micrófono a AVS a través de AWS IoT, el dispositivo deja de escuchar y cambia a un estado de pensamiento. Este estado indica que AVS está procesando la pregunta y determina la mejor respuesta. Mientras el kit se encuentra en este estado, el LED indicador de estado parpadea en color cian y azul a intervalos de 200 ms. Las imágenes siguientes muestran cómo aparece este parpadeo en el kit.



Cuando el dispositivo termina de pensar, comienza a responder. Antes de que el kit comience a hablar, la luz indicadora de estado cambia a un estado de habla. El kit parpadea en color cian y azul a intervalos de 500 ms.

La respuesta de Alexa se reproduce desde el altavoz del kit mientras la luz indicadora de estado parpadea en color cian y azul, Alexa describe las condiciones meteorológicas en función de la ubicación de tu cuenta de consumidor de Alexa. Una vez finalizada la respuesta, la luz indicadora de estado deja de parpadear y se apaga. Esto indica que el kit está inactivo y espera la palabra de activación de Alexa.

Utilizar suAWSy cuentas de desarrollador de Alexa Voice Service para configurar AVS paraAWS IoT

La cuenta NXP preconfigurada solo sirve para evaluar el kit. Al utilizar su propia cuenta, obtendrá los siguientes beneficios.

- Control total de los trabajos e implementaciones sobre el aire (OTA), como actualizaciones de firmware remotas.
- Control sobreAWServicios de .
- Personalización de las habilidades de hogar inteligente.

Para migrar desde la cuenta NXP preconfigurada a su propia cuenta, descargue elGuía de migración de la solución de voz de MCU Alexadesde lasIntroducciónSección sobre de la[Solución basada en MCU EdgeReady para Alexa para IOT](#)(Se ha creado el certificado). Siga los pasos de esta guía.

Note

Para descargar este archivo, necesitas una cuenta NXP.

AWS IoTSDK de dispositivos, SDK móviles yAWS IoTCliente de dispositivos

En esta página se resume elAWS IoTSDK de dispositivos, bibliotecas de código abierto, guías para desarrolladores, aplicaciones de ejemplo y guías de portabilidad para ayudarle a crear soluciones innovadoras de IoT conAWS IoT y las plataformas de hardware que elijas.

Estos SDK se utilizan en su dispositivo IoT. Si estás desarrollando una aplicación IoT para utilizarla en un dispositivo móvil, consulta la[AWS Mobile SDK \(p. 1276\)](#). Si estás desarrollando una aplicación de IoT o un programa del lado del servidor, consulta la[AWS SDK \(p. 73\)](#).

SDK de dispositivos de AWS IoT

Los SDK de dispositivos de AWS IoT contienen bibliotecas de código abierto, guías de desarrolladores con ejemplos y guías de migración para que pueda crear productos o soluciones de IoT innovadores en las plataformas de hardware deseadas.

Estos SDK te ayudan a conectar tus dispositivos IoT aAWS IoT utilizando los protocolos MQTT y WSS.

C++

AWS IoTC++ Device SDK

LaAWS IoTC++ Device SDK permite a los desarrolladores compilar aplicaciones conectadas medianteAWSy laAWS IoTAPI. En concreto, este SDK se diseñó para los dispositivos que no tienen limitación de recursos y requieren características avanzadas, como la puesta en cola de mensajes, la compatibilidad con varios procesos y las características de idioma más actualizadas. Para obtener más información, consulte los siguientes:

- [AWS IoTDevice SDK C++ v2 en GitHub](#)
- [AWS IoTArchivo léame de Device SDK C++ v2](#)
- [AWS IoTEjemplos de Device SDK C++ v2](#)
- [AWS IoTDocumentación de la API de C++ v2 del SDK del dispositivo](#)

Python

AWS IoTDevice SDK for Python

AWS IoT Device SDK for Python permite a los desarrolladores escribir scripts de Python para tener acceso con sus dispositivos a la plataforma de AWS IoT mediante MQTT o MQTT sobre protocolo WebSocket. Conectando sus dispositivos aAWS IoT, los usuarios pueden trabajar de forma segura con el agente de mensajes, las reglas y las sombras de objeto que proporcionaAWS IoT con otrosAWSservicios comoAWS Lambda, Kinesis y Amazon S3 y más.

- [AWS IoT Device SDK for Python v2 en GitHub](#)
- [AWS IoT Device SDK for Python v2 Readme](#)

- [AWS IoTDevice SDK for Python v2](#)
- [AWS IoTDocumentación de Device SDK for Python v2](#)

JavaScript

AWS IoTDevice SDK for JavaScript

El paquete aws-iot-device-sdk.js permite a los desarrolladores escribir aplicaciones JavaScript que tengan acceso a AWS IoT mediante MQTT o MQTT sobre protocolo WebSocket. Se puede utilizar en entornos de Node.js y aplicaciones de navegador. Para obtener más información, consulte los siguientes:

- [AWS IoT Device SDK for JavaScript v2 en GitHub](#)
- [Archivo léame de AWS IoT Device SDK for JavaScript v2](#)
- [AWS IoTEjemplos de Device SDK for JavaScript v2](#)
- [AWS IoTDocumentación de Device SDK for JavaScript v2](#)

Java

AWS IoTDevice SDK para Java

AWS IoT Device SDK for Java permite a los desarrolladores de Java tener acceso a la plataforma de AWS IoT mediante MQTT o MQTT sobre protocolo WebSocket. El SDK es compatible con las sombras. Puede tener acceso a las sombras mediante los métodos GET, UPDATE y DELETE de HTTP. El SDK es también compatible con un modelo de acceso a sombras simplificado, lo que permite a los desarrolladores intercambiar datos con las sombras utilizando únicamente métodos getter y setter, sin tener que serializar ni deserializar documentos JSON. Para obtener más información, consulte los siguientes:

- [AWS IoT Device SDK for Java v2 en GitHub](#)
- [Archivo léame de AWS IoT Device SDK for Java v2](#)
- [AWS IoTDevice SDK for Java v2](#)
- [AWS IoTDocumentación de Device SDK for Java v2](#)

SDK de dispositivos de AWS IoT para Embedded C

Note

Este SDK está diseñado para ser utilizado por desarrolladores de software integrado experimentados.

La AWS IoT Device SDK para Embedded C(C-SDK) es un conjunto de archivos de origen C con licencia de código abierto del MIT, que se puede utilizar en aplicaciones insertadas para establecer conexiones seguras con dispositivos IoT a AWS IoT Core. Incluye un cliente MQTT, JSON Parser y AWS IoTDevice Shadow, AWS IoTTrabajos, AWS IoTAprovisionamiento de flotas y AWS IoT Device DefenderBibliotecas de Este SDK se distribuye en forma de origen y se puede integrar en el firmware cliente junto con el código de aplicación, otras bibliotecas y un sistema operativo (SO) de su elección.

AWS IoT Device SDK para Embedded C generalmente se dirige a dispositivos con limitaciones de recursos que requieren un tiempo de ejecución optimizado del lenguaje C. Puede usar el SDK en cualquier sistema operativo y alojarlo en cualquier tipo de procesador (por ejemplo, MCU y MPU).

Para obtener más información, consulte los siguientes:

- [AWS IoT Device SDK para Embedded C en GitHub](#)

- [AWS IoT Device SDK for Embedded C Readme](#)
- [AWS IoTDevice SDK for Embedded C Samples](#)

AntesAWS IoTVersiones del SDK de dispositivos

Se trata de versiones anteriores de AWS IoTSDK de dispositivo que se han sustituido por las versiones más recientes enumeradas anteriormente. Estos SDKs solo reciben actualizaciones de mantenimiento y actualizaciones de seguridad. No se actualizarán para incluir nuevas funciones y no deben utilizarse en nuevos proyectos.

- [AWS IoT C++ Device SDK en GitHub](#)
- [AWS IoT C++ Device SDK Readme](#)
- [AWS IoT Device SDK for Python v1 on GitHub](#)
- [AWS IoT Device SDK for Python v1 Readme](#)
- [AWS IoT Device SDK for Java on GitHub](#)
- [AWS IoT Device SDK for Java Readme](#)
- [AWS IoT Device SDK for JavaScript on GitHub](#)
- [AWS IoT Device SDK for JavaScript Readme](#)
- [Arduino Yún SDK on GitHub](#)
- [Arduino Yún SDK Readme](#)

AWS Mobile SDK

Los SDK móviles proporcionan soporte específico de la plataforma para desarrolladores de aplicaciones móviles para las API de AWS IoT Coreservicios, comunicación de dispositivos IoT mediante MQTT y las API de otros AWS Servicios de .

Android

AWS Mobile SDK for Android

La AWS Mobile SDK for Android contiene una biblioteca, ejemplos y documentación para que los desarrolladores creen aplicaciones móviles conectadas mediante AWS. Este SDK también permite establecer comunicaciones de dispositivos MQTT y llamar a las API de AWS IoT Core Servicios de . Para obtener más información, consulte los siguientes:

- [AWS Mobile SDK for Android en GitHub](#)
- [AWS Mobile SDK for Android Readme \(Léame\)](#)
- [Ejemplos de AWS Mobile SDK for Android](#)
- [Referencia de la API de AWS Mobile SDK for Android](#)
- [Documentación de referencia de la clase AWSSlotClient](#)

iOS

AWS Mobile SDK for iOS

La AWS Mobile SDK for iOS es un kit de desarrollo de software de código abierto, distribuido con licencia de Apache Open Source. La AWS Mobile SDK for iOS proporciona una biblioteca, ejemplos de código y documentación para ayudar a los desarrolladores a crear aplicaciones móviles conectadas mediante AWS. Este SDK también permite establecer comunicaciones de dispositivos MQTT y llamar a las API de AWS IoT Core Servicios de . Para obtener más información, consulte los siguientes:

- [AWS Mobile SDK for iOS en GitHub](#)
- [AWS Mobile SDK for iOSReadme \(Léame\)](#)
- [Ejemplos de AWS Mobile SDK for iOS](#)
- [Documentos de referencia de clase AWSslot en elAWS Mobile SDK for iOS](#)

AWS IoTClient de dispositivos

La AWS IoTDevice Client proporciona código para ayudar a que el dispositivo se conecte a AWS IoT, realice tareas de aprovisionamiento de flotas, soporte de políticas de seguridad de dispositivos, conéctese mediante túneles seguros y procese trabajos en su dispositivo. Puede instalar este software en su dispositivo para gestionar estas tareas rutinarias del dispositivo y así poder centrarse en su solución específica.

Note

La AWS IoTDevice Client funciona con dispositivos IoT basados en microprocesadores con procesadores x86_64 o ARM y sistemas operativos Linux comunes.

C++

AWS IoTClient de dispositivos

Para obtener más información acerca de AWS IoTDevice Client en C++, consulte lo siguiente:

- [AWS IoTDevice Client in C ++ en GitHub](#)
- [AWS IoTClient de dispositivo en el archivo Léame de C++](#)

Solución de problemas de AWS IoT

Ayúdenos a mejorar este tema

[Háganos saber qué ayudaría a mejorar](#)

La siguiente información puede ayudarle a solucionar problemas comunes en AWS IoT.

Tareas

- [Diagnóstico de problemas de conectividad \(p. 1278\)](#)
- [Diagnóstico de problemas de las reglas \(p. 1281\)](#)
- [Diagnóstico de problemas relacionados con las sombras \(p. 1282\)](#)
- [Diagnosticar problemas con acciones del flujo de entrada de Salesforce IoT \(p. 1283\)](#)
- [Guía para solucionar problemas de indexación de flota \(p. 1284\)](#)
- [Solución de problemas de «Superación del límite de secuencias deAWS cuenta» \(p. 1286\)](#)
- [Guía para solucionar problemas de AWS IoT Device Defender \(p. 1286\)](#)
- [AWS IoTGuía para solucionar problemas de Device Advisor \(p. 1290\)](#)
- [Solución de problemas de desconexión de flota de dispositivos \(p. 1291\)](#)
- [Errores de AWS IoT \(p. 1292\)](#)

Diagnóstico de problemas de conectividad

Ayúdenos a mejorar este tema

[Háganos saber qué ayudaría a mejorar](#)

Una conexión exitosa conAWS IoT requiere:

- Una conexión válida
- Un certificado válido y activo
- Una política que permite la conexión y el funcionamiento deseados

Conexión

¿Cómo encuentro el punto final correcto?

- El valor de `endpointAddress` devuelto por `aws iot describe-endpoint --endpoint-type iot:Data-ATS`
 - o bien
- El valor de `domainName` devuelto por `aws iot describe-domain-configuration --domain-configuration-name "domain_configuration_name"`

¿Cómo puedo encontrar el valor correcto de Indicación de nombre de servidor (SNI)?

El valor de SNI correcto es el `endpointAddress` devuelto por el `describe-endpoint` o `describe-domain-configuration` commands. Es la misma dirección que el punto de enlace en el paso anterior.

¿Cómo soluciono un problema de conectividad que persiste?

Puede usar AWS Device Advisor para ayudar a solucionar problemas. Las pruebas predefinidas de Device Advisor le ayudan a validar el software de su dispositivo con respecto a las prácticas recomendadas para el uso de [TLS](#), [MQTT](#), [AWS IoT Sombra de dispositivos](#), y [AWS IoT Trabajos](#).

Aquí hay un enlace a la existente [Device Advisor](#) contenido.

Autenticación

Los dispositivos deben ser autenticados ([p. 297](#)) para conectar a AWS IoT. Puntos de enlace de . Para dispositivos que utilizan [Certificados de cliente X.509](#) ([p. 298](#)) para la autenticación, los certificados deben registrarse con AWS IoT y mantenerse activos.

¿Cómo autentican mis dispositivos los puntos de enlace de AWS IoT?

Agregue el certificado de entidad de certificación de AWS IoT al almacén de confianza de su cliente. Consulte la documentación de [Autenticación de servidor AWS IoT Core](#) y siga los enlaces para descargar el certificado de CA correspondiente.

Qué se comprueba cuando un dispositivo se conecta a AWS IoT?

Cuando un dispositivo intenta conectarse a AWS IoT:

1. AWS IoT busca un certificado y un valor de Indicación de nombre de servidor (SNI) válidos.
2. AWS IoT comprueba que el certificado utilizado está registrado en el AWS IoT Cuenta y que se ha activado.
3. Cuando un dispositivo intenta realizar cualquier acción en AWS IoT, por ejemplo, para suscribirse o publicar un mensaje, se comprueba la directiva adjunta al certificado que utilizó para conectarse para confirmar que el dispositivo está autorizado a realizar esa acción.

¿Cómo puedo validar un certificado configurado correctamente?

Ejecute el comando `s_client` de OpenSSL para probar una conexión con el punto de enlace de AWS IoT:

```
openssl s_client -connect custom_endpoint.iot.aws-region.amazonaws.com:8443 -  
CAfile CA.pem -cert cert.pem -key privateKey.pem
```

Para obtener más información sobre el uso de `openssl s_client`, consulte [Documentación de OpenSSL s_client](#).

¿Cómo puedo comprobar el estado de un certificado de?

- Enumeración de los certificados

Si no conoce el identificador del certificado, puede ver el estado de todos sus certificados mediante el comando `aws iot list-certificates`.

- Mostrar los detalles de un certificado

Si conoce el identificador del certificado, este comando le muestra información más detallada sobre el certificado.

```
aws iot describe-certificate --certificate-id "certificateId"
```

- Revise el certificado en el AWS IoT Consola

En el navegador [AWS IoT Consola](#), en el menú de la izquierda, elija Seguridad y luego elija Certificados.

Elija el certificado que está utilizando para conectarse de la lista y abrir su página de detalles.

En la página de detalles del certificado, puede ver su estado actual.

El estado del certificado se puede cambiar mediante laActionsen la esquina superior derecha de la página de detalles.

Autorización

AWS IoTUso de recursos de[Políticas de AWS IoT Core \(p. 333\)](#)autorizar a esos recursos a llevar a cabo[Acciones de \(p. 334\)](#). Para que se autorice una acción, elAWS IoTlos recursos deben tener adjunto un documento de política que otorgue permiso para llevar a cabo esa acción.

He recibido una respuesta PUBNACK o SUBNACK del agente. ¿Qué tengo que hacer?

Asegúrese de que el certificado que utilice para llamar a AWS IoT tenga una política asociada. De forma predeterminada, todas las operaciones de publicación o suscripción se deniegan.

Asegúrese de que la política adjunta autorice el[Acciones de \(p. 334\)](#)está intentando actuar.

Asegúrese de que la política adjunta autorice el[recursos \(p. 336\)](#)que intentan realizar las acciones autorizadas.

Tengo unAUTHORIZATION_FAILUREEntrada en mis registros.

Asegúrese de que el certificado que utilice para llamar a AWS IoT tenga una política asociada. De forma predeterminada, todas las operaciones de publicación o suscripción se deniegan.

Asegúrese de que la política adjunta autorice el[Acciones de \(p. 334\)](#)está intentando actuar.

Asegúrese de que la política adjunta autorice el[recursos \(p. 336\)](#)que intentan realizar las acciones autorizadas.

¿Cómo verifico lo que autoriza la política?

En el navegador[AWS IoTconsola](#), en el menú de la izquierda, elijaSeguridady luego elijaCertificados.

Elija el certificado que está utilizando para conectarse de la lista y abrir su página de detalles.

En la página de detalles del certificado, puede ver su estado actual.

En el menú izquierdo de la página de detalles del certificado, elijaPolíticasPara ver las políticas asociadas al certificado

Elija la política deseada para ver su página de detalles.

En la página de detalles de la política, revise lasDocumento de políticapara ver qué autoriza.

ElegirEdición del documento de políticapara realizar cambios en el documento de política.

Seguridad e identidad

Cuando proporciona los certificados de servidor paraAWS IoTconfiguración de dominio personalizada, los certificados tienen un máximo de cuatro nombres de dominio.

Para obtener más información, consulte [Puntos de enlace y cuotas de AWS IoT Core](#).

Diagnóstico de problemas de las reglas

Ayúdenos a mejorar este tema

[Háganos saber qué ayudaría a mejorarla](#)

En esta sección se describen algunas de las cosas que hay que comprobar cuando se produce un problema con la regla.

Configuración de CloudWatch Logs para la solución de problemas

La mejor forma de solucionar problemas con las reglas es utilizar CloudWatch Logs. Cuando habilita CloudWatch Logs para AWS IoT, puede ver qué reglas se activan, así como su éxito o fracaso. También obtiene información sobre si las condiciones de la cláusula WHERE coinciden. Para obtener más información, consulte [Monitor AWS IoT con CloudWatch Registros \(p. 449\)](#).

El problema más habitual de las reglas es la autorización. Los registros se muestran si el rol no está autorizado a realizar operaciones AssumeRole en el recurso. A continuación hay un log de ejemplo generado por el [registro detallado \(p. 431\)](#):

```
{  
    "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/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: AKQJ987654321AKQJ123456789AKQJ987654321AKQJ987654321)"  
}
```

A continuación hay un log de ejemplo similar generado por el [registro global \(p. 429\)](#):

```
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 obtener más información, consulte [the section called “Visualización de AWS IoT inicio de sesión en el CloudWatch consola” \(p. 449\)](#).

Diagnóstico de servicios externos

El usuario final controla los servicios externos. Antes de ejecutar la regla, asegúrese de que los servicios externos que ha vinculado a su regla estén configurados y tengan suficientes unidades de capacidad y procesamiento para su aplicación.

Diagnóstico de problemas de SQL

Si la consulta SQL no devuelve los datos que espera:

- Revise los registros en busca de mensajes de error.
- Confirme que la sintaxis SQL coincide con el documento JSON del mensaje.

Revise los nombres de objetos y propiedades utilizados en la consulta con los utilizados en el documento JSON de la carga útil del mensaje del tema. Para obtener más información sobre el formato JSON en las consultas SQL, consulte [Extensiones JSON \(p. 620\)](#).

- Compruebe si los nombres de objetos o propiedades JSON incluyen caracteres reservados o numéricos.

Para obtener más información sobre los caracteres reservados en las referencias de objetos JSON en consultas SQL, consulte [Extensiones JSON \(p. 620\)](#).

Diagnóstico de problemas relacionados con las sombras

Ayúdenos a mejorar este tema

[Háganos saber qué ayudaría a mejorar](#)

Diagnóstico de sombras

Problema	Directrices para solucionar problemas
El documento de la sombra de un dispositivo se rechaza con <code>Invalid JSON document</code> .	Si no está familiarizado con JSON, modifique los ejemplos proporcionados en esta guía y adáptelos a sus necesidades. Para obtener más información, consulte Ejemplos de documento de sombra (p. 665) .
He enviado un JSON correcto, pero no se almacena o solo se almacena parcialmente en el documento de sombra del dispositivo.	Compruebe que ha seguido las directrices de formato JSON. Solo se almacenarán los campos JSON de las secciones <code>desired</code> y <code>reported</code> . No se tendrá en cuenta el contenido JSON (aunque sea formalmente correcto) que no esté en estas secciones.
He recibido un error que indica que la sombra del dispositivo supera el tamaño máximo permitido.	La sombra del dispositivo admite únicamente 8 KB de datos. Intente acortar los nombres de los campos que están dentro del documento JSON o cree más sombras generando más objetos.

Problema	Directrices para solucionar problemas
	Un dispositivo puede tener asociado un número ilimitado de objetos o sombras. El único requisito es que cada nombre de objeto debe ser único en la cuenta.
Cuando recibo la sombra de un dispositivo, esta supera los 8 KB. ¿Por qué pasa esto?	En la recepción, el servicio de AWS IoT agrega metadatos a la sombra del dispositivo. El servicio incluye estos datos en su respuesta, pero no cuentan para el límite de 8 KB. Para calcular el límite, solo se tienen en cuenta los datos de estado <code>desired</code> y <code>reported</code> del documento de estado enviado a la sombra del dispositivo.
Mi solicitud se ha rechazado porque la versión es incorrecta. ¿Qué tengo que hacer?	Realice una operación GET para sincronizarse con la última versión del documento de estado. Al usar MQTT, suscríbase al tema <code>/update/accepted</code> para recibir notificaciones sobre cambios de estado y la última versión del documento JSON.
La marca de tiempo está desajustada en varios segundos.	El servicio AWS IoT actualiza la marca de tiempo de algunos campos individuales y de todo el documento JSON cuando los recibe; la marca de tiempo también se actualiza cuando el documento de estado se publica en el mensaje <code>./update/accepted</code> y <code>./update/delta</code> . Los mensajes pueden retrasarse en la red, lo que puede provocar una demora de varios segundos en la marca de tiempo.
Mi dispositivo puede publicar en los temas de sombra correspondientes y suscribirse a ellos, pero cuando intento actualizar el documento de sombra mediante la API de REST de HTTP, recibo un mensaje HTTP 403.	Compruebe que haya creado políticas en IAM que permitan a sus credenciales tener acceso a estos temas y a su acción correspondiente (UPDATE/GET/DELETE). Las políticas de IAM y las de certificado son independientes.
Otros problemas.	El servicio Device Shadow registra errores en CloudWatch Logs. Para identificar los problemas de configuración y de dispositivo, active CloudWatch Logs y consulte los logs para obtener información de depuración.

Diagnosticar problemas con acciones del flujo de entrada de Salesforce IoT

Ayúdenos a mejorar este tema

[Háganos saber qué ayudaría a mejorarla](#)

Registro de seguimiento de ejecución

¿Cómo puedo ver el registro de seguimiento de ejecución de una acción de Salesforce?

Consulte la sección [MonitorAWS IoT con CloudWatch Registros \(p. 449\)](#). Una vez que haya activado los registros, podrá ver el seguimiento de la ejecución de la acción de Salesforce.

Éxito y error de una acción

¿Cómo puedo saber que los mensajes se han enviado correctamente a un flujo de entrada de Salesforce IoT?

Consulte los logs generados por la ejecución de la acción de Salesforce en CloudWatch Logs. Si aparece `Action executed successfully`, eso significa que el motor de reglas de AWS IoT recibió la confirmación de Salesforce IoT de que el mensaje se envió correctamente al flujo de entrada de destino.

Si tiene problemas con la plataforma de Salesforce IoT, póngase en contacto con la ayuda de Salesforce IoT.

¿Qué hago si los mensajes no se han enviado correctamente a un flujo de entrada de Salesforce IoT?

Consulte los logs generados por la ejecución de la acción de Salesforce en CloudWatch Logs. En función de la entrada de registro, puede realizar las siguientes operaciones:

`Failed to locate the host`

Compruebe que el parámetro `url` de la acción es correcto y que el flujo de entrada de Salesforce IoT existe.

`Received Internal Server Error from Salesforce`

Reintentar. Si el problema continúa, póngase en contacto con el equipo de soporte de Salesforce IoT.

`Received Bad Request Exception from Salesforce`

Compruebe si la carga que envía presenta errores.

`Received Unsupported Media Type Exception from Salesforce`

Salesforce IoT no admite una carga binaria en este momento. Compruebe que está enviando una carga JSON.

`Received Unauthorized Exception from Salesforce`

Compruebe que el parámetro `token` de la acción es correcto y que su token sigue siendo válido.

`Received Not Found Exception from Salesforce`

Compruebe que el parámetro `url` de la acción es correcto y que el flujo de entrada de Salesforce IoT existe.

Si recibe un error que no aparece aquí, póngase en contacto con AWS IoT Support.

Guía para solucionar problemas de indexación de flota

Ayúdenos a mejorar este tema

[Háganos saber qué ayudaría a mejorar](#)

Solución de problemas de consultas de agregación en el servicio de indexación de flotas

Si surgen errores de discordancia de tipo, puede utilizar CloudWatch Logs para solucionar el problema. CloudWatch Logs deben estar habilitado para que el servicio de indexación de flotas pueda crear los logs. Para obtener más información, consulte [MonitorAWS IoT con CloudWatch Registros \(p. 449\)](#).

Cuando realiza consultas de agregación en campos no administrados, solo puede especificar un campo definido en el argumento `customFields` pasado a `UpdateIndexingConfiguration` o `update-indexing-configuration`. Si el valor del campo no coincide con el tipo de datos del campo configurado, este valor se omite al realizar una consulta de agregación.

El servicio de indexación de flotas envía un registro de errores a CloudWatch Logs cuando un campo no se puede indexar debido a un tipo no coincidente. El registro de errores contiene el nombre del campo, el valor que no se pudo convertir y el nombre de objeto del dispositivo. A continuación, se muestra un ejemplo de registro de error.

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

Si un dispositivo se ha desconectado durante aproximadamente una hora, el valor `timestamp` del estado de conectividad podría no aparecer. Para las sesiones persistentes, el valor podría no aparecer después de que un cliente se haya desconectado durante un periodo mayor que el tiempo de vida (TTL) configurado para la sesión persistente. Los datos de estado de conectividad solo se indexan para las conexiones donde el ID de cliente tiene un nombre de objeto coincidente. (El ID de cliente es el valor que se utiliza para conectar un dispositivo a AWS IoT Core).

Solución de problemas de métricas de flota

No se puede crear una métrica de flota

No se admite la degradación de fuentes de datos mediante la actualización de la configuración de indexación de flotas.

Si intenta crear una métrica de flota con fuentes de datos degradadas (por ejemplo, anteriormente los orígenes de datos eran datos de registro, datos de sombra y datos de conectividad de dispositivos, y ahora los orígenes de datos son datos de registro y datos sombra y sin datos de conectividad de dispositivos), verá errores y no podrá crear una métrica de flota.

No se admite la modificación de los campos personalizados utilizados por las métricas de flotas existentes.

No se pueden ver los puntos de datos en CloudWatch

Si puede crear una métrica de flota pero no puede ver puntos de datos en CloudWatch, es probable que no tenga nada que cumpla los criterios de cadena de consulta.

Consulte este ejemplo de comando de cómo crear una métrica de flota:

```
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
```

Si no tiene nada que cumpla los criterios de cadena de consulta--query-string
"thingName:TempSensor* AND attributes.temperature>80":

- convalues=count, podrá crear una métrica de flota y habrá puntos de datos que mostrar en CloudWatch. Los puntos de datos del valorcountes siempre 0.
- convaluesdistintos decount, podrá crear una métrica de flota pero no verá la métrica de flota en CloudWatch y no habrá puntos de datos que mostrar en CloudWatch.

Solución de problemas de «Superación del límite de secuencias de AWS cuenta»

Ayúdenos a mejorar este tema

[Háganos saber qué ayudaría a mejorar](#)

Si aparece "Error: You have exceeded the limit for the number of streams in your AWS account.", puede limpiar las secuencias no utilizadas de su cuenta en lugar de solicitar un aumento del límite.

Para limpiar un flujo sin usar que ha creado mediante el AWS CLI o SDK de:

```
aws iot delete-stream -stream-id value
```

Para obtener más detalles, consulte [delete-stream](#).

Note

Puede utilizar el `list-streams` para encontrar los identificadores de secuencias.

Guía para solucionar problemas de AWS IoT Device Defender

Ayúdenos a mejorar este tema

[Háganos saber qué ayudaría a mejorar](#)

Generales

Q: ¿Hay algún requisito previo para usar AWS IoT Device Defender?

A: Si desea utilizar métricas registradas por el dispositivo, primero debe implementar un agente en su AWS IoT dispositivos conectados o puertas de enlace de dispositivos. Los dispositivos deben proporcionar un identificador de cliente o nombre de objeto coherentes.

Auditoría

Q: Permití una comprobación y mi auditoría ha estado mostrando «En curso» durante mucho tiempo. ¿Hay algún problema? ¿Cuándo recibiré los resultados?

A: Cuando se habilita una comprobación, la recopilación de datos comienza inmediatamente. Sin embargo, si la cuenta tiene una gran cantidad de datos que recopilar (por ejemplo, certificados, objetos o políticas), es posible que los resultados de la comprobación no estén disponibles durante cierto tiempo después de haberse habilitado.

Detect

Q: ¿Cómo puedo conocer los umbrales que se establecen en unAWS IoT Device Defendercomportamiento del perfil de seguridad?

A: Comience por crear un comportamiento del perfil de seguridad con umbrales bajos y asócielo a un grupo de objetos que conste de un conjunto representativo de dispositivos. Puede utilizar AWS IoT Device Defender para ver las métricas actuales y después ajustar el comportamiento de los umbrales para que coincidan con su caso de uso.

Q: He creado un comportamiento, pero no activa una vulneración cuando la espero. ¿Cómo debo solucionarlo?

A: Cuando define un comportamiento, especifica la forma en que espera que el dispositivo se comporte con normalidad. Por ejemplo, si tiene una cámara de seguridad que se conecta exclusivamente a un servidor central en el puerto TCP 8888, no espere que se realicen otras conexiones. Para recibir una alerta si la cámara hace una conexión en otro puerto, puede definir un comportamiento como este:

```
{  
  "name": "Listening TCP Ports",  
  "metric": "aws:listening-tcp-ports",  
  "criteria": {  
    "comparisonOperator": "in-port-set",  
    "value": {  
      "ports": [ 8888 ]  
    }  
  }  
}
```

Si la cámara realiza una conexión TCP en el puerto TCP 443, el comportamiento del dispositivo se infringiría y se activaría una alerta.

Q: Se está vulnerando uno o más de mis comportamientos. ¿Cómo elimino la vulneración?

A: Las alarmas se desactivan después de que el dispositivo retoma un comportamiento esperado, tal y como se define en los perfiles de comportamiento. Los perfiles de comportamiento se evalúan al recibir los datos de las métricas de su dispositivo. Si el dispositivo no publica ninguna métrica durante más de dos días, el evento de vulneración se establece en `alarm-invalidated` automáticamente.

Q: Eliminé un comportamiento que generaba una vulneración; ¿cómo puedo detener las alertas?

A: Al eliminar un comportamiento se detienen todas las vulneraciones y alertas futuras de dicho comportamiento. Las alertas anteriores deben eliminarse del mecanismo de notificación. Cuando se elimina un comportamiento, el registro de vulneraciones de ese comportamiento se conserva durante el mismo período de tiempo que todas las demás vulneraciones en su cuenta.

Métricas de dispositivo

Q: Estoy enviando informes de métricas que sé que vulneran mis comportamientos, pero no se activa ninguna vulneración. ¿Por qué?

A: Compruebe que sus informes de métricas se estén aceptando suscribiéndose a los siguientes temas de MQTT:

```
$aws/things/THING_NAME/defender/metrics/FORMAT/rejected  
$aws/things/THING_NAME/defender/metrics/FORMAT/accepted
```

THING_NAME es el nombre de la cosa que informa la métrica y FORMATEs «JSON» o «CBOR» según el formato del informe de métricas que envía el objeto.

Una vez que se haya suscrito, debe recibir mensajes sobre estos temas para cada informe de métrica enviado. Un mensaje rejected indica que hubo un problema al analizar el informe de métrica. Se incluye un mensaje de error en la carga del mensaje para ayudarle a corregir cualquier error en su informe de métrica. Unaccepted indica que se ha analizado correctamente el informe de métrica.

Q: ¿Qué sucede si envío una métrica vacía en mi informe de métrica?

A: Una lista vacía de puertos o direcciones IP se considera siempre que está en conformidad con el comportamiento correspondiente. Si el comportamiento correspondiente supusiera una vulneración, la vulneración se eliminaría.

Q: ¿Por qué los informes de métricas de mi dispositivo contienen mensajes para dispositivos que no están en la AWS IoT Registro?

Si tiene uno o varios perfiles de seguridad asociados a todos los objetos o a todos los objetos no registrados, AWS IoT Device Defender incluye métricas de los objetos no registrados. Si desea excluir las métricas de objetos no registrados, puede asociar los perfiles a todos los dispositivos registrados en lugar de a todos los dispositivos.

Q: No veo mensajes de uno o varios dispositivos no registrados a pesar de que he aplicado un perfil de seguridad a todos los dispositivos no registrados o a todos los dispositivos. ¿Cómo puedo solucionarlo?

Compruebe que está enviando un informe de métricas bien formado utilizando uno de los formatos compatibles. Para obtener información, consulte [Especificación de documentos de métricas de dispositivos \(p. 1012\)](#). Compruebe que los dispositivos no registrados utilizan un identificador de cliente o nombre de objeto coherentes. Si el nombre del objeto contiene caracteres de control o tiene más de 128 bytes de caracteres con codificación UTF-8, los mensajes registrados por los dispositivos se rechazan.

Q: ¿Qué sucede si un dispositivo no registrado se añade al registro o si un dispositivo registrado deja de estarlo?

A: Si un dispositivo se añade o se elimina del registro:

- Verá dos vulneraciones independientes para el dispositivo (una en su nombre de objeto registrado y otra en su identidad no registrada) si se siguen publicando métricas para las vulneraciones. Las vulneraciones activas de la identidad antigua dejan de aparecer al cabo de dos días, pero están disponibles en el historial de vulneraciones durante un máximo de 14 días.

Q: ¿Qué valor debo proporcionar en el campo de ID del informe de métricas de mi dispositivo?

A: Utilice un valor único para cada informe de métrica, expresado como un número entero positivo. Una práctica común es utilizar una [marca de tiempo Epoch de Unix](#).

Q: ¿Debo crear una conexión con MQTT dedicada para AWS IoT Device Defender? ¿Métricas de?

A: No se requiere una conexión con MQTT independiente.

Q: ¿Qué ID de cliente debería usar al conectarme para publicar métricas de dispositivos?

Para dispositivos (objetos) que estén el registro de AWS IoT registro, utilice el nombre del objeto registrado. Para los productos que no estén en el registro de AWS IoT, utilice un identificador coherente al conectarse a AWS IoT. Esta práctica le permite crear una correspondencia entre las vulneraciones y el nombre de objeto.

Q: ¿Puedo publicar métricas para un dispositivo con un ID de cliente diferente?

Es posible publicar métricas en nombre de otro objeto. Para ello, publique las métricas en el tema de AWS IoT Device Defender reservado para dicho dispositivo. Por ejemplo, Thing-1 desea publicar métricas de sí mismo y también en nombre de Thing-2. Thing-1 recopila sus propias métricas y las publica en el tema de MQTT:

```
$aws/things/Thing-1/defender/metrics/json
```

Thing-1 obtiene las métricas de Thing-2 y publica dichas métricas en el tema de MQTT:

```
$aws/things/Thing-2/defender/metrics/json
```

Q: ¿Cuántos comportamientos y perfiles de seguridad puedo tener en mi cuenta?

A: Consulte [AWS IoT Device Defender Cuotas y puntos de enlace de](#).

Q: ¿Qué aspecto tiene un rol de destino prototípico para un destino de alerta?

A: Un rol que permite AWS IoT Device Defender para publicar alertas en un destino de alerta (tema SNS) requiere dos cosas:

- Una relación de confianza que especifique `iot.amazonaws.com` como la entidad de confianza y
- Una política asociada que conceda a AWS IoT permiso para publicar en un tema de SNS especificado. Por ejemplo:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "sns:Publish",
            "Resource": "<sns-topic-arn>"
        }
    ]
}
```

- Si el tema SNS utilizado para publicar alertas es un tema cifrado, junto con el permiso para publicar en el tema SNS, AWS IoT debe concederse dos permisos más. Por ejemplo:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "sns:Publish",
                "kms:Decrypt",
                "kms:GenerateDataKey"
            ],
            "Resource": "<sns-topic-arn>"
        }
    ]
}
```

Q: Envío de mi informe de métricas con un tipo de métrica personalizada `number` falla con el mensaje de error `Malformed metrics report`. ¿Por qué?

A: El tipo `number` solo toma un solo valor de métrica como entrada, pero al enviar el valor de métricas en el informe `DeviceMetrics`, debe pasarse como una matriz con un único valor. Asegúrese de enviar el valor de la métrica como matriz.

Carga útil del error:

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

Mensaje de error:

```
{"thingName": "myThing", "status": "REJECTED", "statusDetails": {"ErrorCode": "InvalidPayload", "ErrorMessage": "Malformed metrics report"}, "timestamp": 1635802047699}
```

Carga útil sin errores:

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

Response: (Respuesta:)

```
{"thingName": "myThing", "12334567": 1635800375, "status": "ACCEPTED", "timestamp": 1635801636023}
```

AWS IoTGuía para solucionar problemas de Device Advisor

Ayúdenos a mejorar este tema

[Háganos saber qué ayudaría a mejorarla](#)

Generales

Q: ¿Puedo ejecutar varios conjuntos de pruebas en paralelo?

A: Sí. Device Advisor ahora admite la ejecución de varios conjuntos de pruebas en distintos dispositivos mediante un extremo de nivel de dispositivo. Si utiliza el endpoint de nivel de cuenta, puede ejecutar un conjunto a la vez porque hay un punto de enlace de Device Advisor disponible por cuenta. Para obtener más información, consulte [Configuración del dispositivo](#).

Q: Vi desde mi dispositivo que Device Advisor denegó la conexión TLS. ¿Se espera esto?

A: Sí. Device Advisor niega la conexión TLS antes y después de cada prueba. Recomendamos a los usuarios que implementen un mecanismo de reintento de dispositivos para tener una experiencia de prueba totalmente automatizada con Device Advisor. Si ejecuta un conjunto de pruebas con más de un caso de prueba, por ejemplo TLS connect, MQTT connect y MQTT publish, le recomendamos que tenga un mecanismo creado para su dispositivo. El mecanismo puede intentar conectarse a nuestro punto final de prueba cada 5 segundos durante un minuto o dos. De esta forma, puede ejecutar varios casos de prueba en secuencia de forma automatizada.

Q: ¿Puedo obtener un historial de llamadas a la API de Device Advisor realizadas en mi cuenta a los fines de realizar tareas de análisis de seguridad y solución de problemas operativos?

A: Sí. Para recibir un historial de llamadas a la API de Device Advisor realizadas en su cuenta, simplemente active CloudTrail en la AWS IoT Management Console y filtre el origen de eventos que se va a [iotdeviceadvisor.amazonaws.com](#).

Q: ¿Cómo veo los registros de Device Advisor en CloudWatch?

A: Los registros generados durante la ejecución de un conjunto de pruebas se cargan en CloudWatch si agrega la política requerida (por ejemplo, CloudWatchFullAccess) a su rol de servicio ([consulte Configuración \(p. 1058\)](#)). Si hay al menos un caso de prueba en el conjunto de pruebas, se crea un grupo de registros «AWS/IOT/DeviceAdvisor/\$testSuiteID» con dos flujos de registro. Un flujo es el «\$testRunid» e incluye registros de las acciones realizadas antes y después de ejecutar los casos de prueba en el conjunto de pruebas, como los pasos de configuración y limpieza. El otro flujo de registro es «\$SuiterUnid_\$testRunid», que es específico de un conjunto de pruebas ejecutado. Eventos enviados desde dispositivos y AWS IoT Core se registrarán en este flujo de registro.

Q: ¿Cuál es el propósito de la función de permiso de dispositivo?

A: Device Advisor se encuentra entre su dispositivo de prueba y AWS IoT Core para simular escenarios de prueba. Acepta conexiones y mensajes de sus dispositivos de prueba y los reenvía a AWS IoT Core asumiendo la función de permiso de dispositivo e iniciando una conexión en su nombre. Es importante asegurarse de que los permisos de rol de dispositivo sean los mismos que los del certificado que utiliza para ejecutar pruebas. AWS IoT las directivas de certificados no se aplican cuando Device Advisor inicia una conexión a AWS IoT Core en su nombre mediante la función de permisos de dispositivo. Sin embargo, se aplican los permisos de la función de permisos de dispositivo que ha establecido.

Q: ¿En qué regiones admite Device Advisor?

A: Device Advisor es compatible con las regiones us-east-1, us-west-2, ap-northeast-1 y eu-west-1.

Q: ¿Por qué veo resultados incoherentes?

A: Una de las principales causas de resultados incoherentes es establecer una prueba `EXECUTION_TIMEOUT` a un valor demasiado bajo. Para obtener más información acerca de las recomendaciones y las predeterminadas `EXECUTION_TIMEOUT` valores, consulte [Casos de prueba de Device Advisor](#).

Q: ¿Qué protocolo MQTT admite Device Advisor?

A: Device Advisor admite MQTT con certificados de cliente X509.

Q: ¿Qué pasa si mi caso de prueba falla con un mensaje de tiempo de espera de ejecución aunque he intentado conectar mi dispositivo al punto final de prueba?

A: Valide todos los pasos descritos en [Crear un rol de IAM para utilizarlo como rol de dispositivo](#). Si la prueba sigue fallando, podría ser que el dispositivo no esté enviando la extensión de indicación de nombre del servidor (SNI) correcta, necesaria para que Device Advisor funcione. El valor SNI correcto es la dirección del punto final devuelta al seguir el [Configuración de la sección del dispositivo](#). AWS IoT también requiere que los dispositivos envíen la extensión Indicación de nombre de servidor (SNI) al protocolo Transport Layer Security (TLS). Para obtener más información, consulte [Seguridad de transporte en AWS IoT](#).

Solución de problemas de desconexión de flota de dispositivos

Ayúdenos a mejorar este tema

Háganos saber qué ayudaría a mejorar

AWS IoT Las desconexiones de la flota de dispositivos pueden producirse por varios motivos. En este artículo se explica cómo diagnosticar un motivo de desconexión y cómo manejar las desconexiones causadas por el mantenimiento regular de AWS IoT servicio o límite de limitación.

Para diagnosticar el motivo de desconexión

Puedes consultar la [AWS IoT Logs V2](#) grupo de registros de [CloudWatch](#) para identificar el motivo de desconexión en el `disconnectReason` de la entrada del registro.

También puede utilizar AWS IoT's [Eventos del ciclo de vida](#) para identificar el motivo de desconexión. Si te has suscrito a `evento de desconexión del ciclo de vida` (`$aws/events/presence/disconnected/`), recibirás una notificación de AWS IoT cuando se produce la desconexión. Puede identificar el motivo de desconexión en el `disconnectReason` campo de la notificación.

Para obtener más información, consulte [CloudWatch AWS IoT Entradas de registro de eventos del ciclo de vida](#).

Para solucionar problemas de desconexiones debidas a AWS IoT mantenimiento del servicio

Desconexiones causadas por AWS IoT mantenimiento del servicio se registran como `SERVER_INITIATED_DISCONNECT` en AWS IoT evento del ciclo de vida y CloudWatch. Para gestionar estas desconexiones, ajusta la configuración del lado del cliente para asegurarte de que los dispositivos se puedan volver a conectar automáticamente a la AWS IoT plataforma.

Para solucionar problemas de desconexiones debido a un límite de estrangulación

Las desconexiones causadas por un límite de restricción se registran como `THROTTLED` en AWS IoT evento del ciclo de vida y CloudWatch. Para gestionar estas desconexiones, puedes solicitar [aumenta el límite del agente de mensajes](#) a medida que crece el recuento de dispositivos.

Para obtener más información, consulte [AWS IoT Agente de mensajes principales](#).

Errores de AWS IoT

Ayúdenos a mejorar este tema

[Háganos saber qué ayudaría a mejorar](#)

En esta sección se presentan los códigos de error que envía AWS IoT.

Códigos de error del agente de mensajes

Código de error	Descripción del error
400	Solicitud errónea.
401	Sin autorización.
403	prohibido.
503	Servicio no disponible.

Identidad y códigos de error de seguridad

Código de error	Descripción del error
401	Sin autorización.

Códigos de error de sombra de dispositivo

Código de error	Descripción del error
400	Solicitud errónea.
401	Sin autorización.
403	prohibido.
404	No encontrado.
409	Conflicto.
413	Solicitud demasiado grande.
422	No se pudo procesar una solicitud.
429	Demasiadas solicitudes.
500	Error interno.
503	Servicio no disponible.

AWS IoT Cuotas de

Para AWS IoT Core información de cuotas, consulte [AWS IoT Core Cuotas y puntos de enlace de en la AWS Referencia general de.](#)

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